KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK–ESAT
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Polynomials in Public Key Cryptography

Promotor:
Prof. Dr. ir. Bart Preneel

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Christopher WOLF

November 2005

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK–ESAT
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Polynomials in Public Key Cryptography

Jury:
Prof. Dr. ir. Guido De Roeck, voorzitter
Prof. Dr. ir. Bart Preneel, promotor
Prof. Dr. ir. André Barbé
Prof. Dr. Jacques Patarin (Université de Versailles)
Prof. Dr. ir. Marc Van Barel
Prof. Dr. ir. Joos Vandewalle
Prof. Dr. ir. Jan Willems

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Christopher WOLF

U.D.C. 681.3*E3

November 2005

D/2005/7515/83

ISBN 90-5682-649-2

In Vielfalt geeint
United in diversity
Unis dans la diversité
In varietate concordia
Unuiĝinte en diverseco
Eenheid in verscheidenheid

Acknowledgements

It is my pleasure to thank many people for helping me to realize this Ph.D. and for making my time here in Leuven much more enjoyable.

First, I want to thank Prof. Bart Preneel for agreeing to be supervisor of this thesis — and finding the money to fund all my trips.

I am also very grateful to Prof. André Barbé, Prof. Jacques Patarin, Prof. Marc Van Barel, and Prof. Joos Vandewalle for serving as jury members. In particular, I want to acknowledge their critical proof reading or the fruitful discussions I had with them in the past. In addition, I want to thank Prof. Jan Willems for agreeing to serve as a jury-member on such short notice and Prof. Guido De Roeck for chairing the jury.

Special thanks go to An Braeken for our successful collaboration, and to Jasper Scholten for patiently answering my "stupid" mathematical questions. Péla Noë and Elvira Wouters deserve a big thank for their support with administrative matters. In addition, I want to thank my current and past COSIC colleges for creating such a nice working atmosphere. This includes in particular Alex Biryukov, Antoon Bosselaers, Christophe De Canniere, Danny De Cock, Thomas Herlea, Elke De Mulder, Klaus Kursawe, Joseph Lano, Gregory Neven, Michael Quisquater, Dries Schellekens, and Brecht Wyseur.

Moreover, I want to thank my former and current house mates Jana Thiers, Els Van Moorhem, Ann Vilette, David Hellin, and Mark Schaerlaekens for their warm welcome, for being so patient with my Dutch ("Bedankt!"), and for gently reminding me from time to time that there is a world outside the university.

Last, but definitely not least, I want to thank my partner Magalie Derhille for the continuous support and fighting off polynomials during our, far too rare, weekends and her family for their heartily welcome. My best thoughts are extended to my parents and my brother, Christoph, Renate, and Stephan Wolf: without their initial support and encouragement, this thesis would not even have been started.

Christopher Wolf
Leuven, November 2005

Abstract

This thesis gives an overview of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomial equations and their use in public key cryptography.

In the first chapter, some general terms of cryptography are introduced. In particular, the need for public key cryptography and alternative schemes is motivated, *i.e.*, systems which neither use factoring (like RSA, Rivest-Shamir-Adleman) nor the discrete logarithm (like ECC, elliptic curve cryptography).

This is followed by a brief introduction of finite fields and a general discussion about $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems of equations and ways of representing them. In this context, affine transformations and their representations are also discussed. After these tools are introduced, they are used to show how $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations can be used for signature and encryption applications. In addition, the problem of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomial equations is put into perspective and a link with the theory of $\mathcal{NP}$-completeness is established. The second chapter concludes with the two related problems *isomorphism of polynomials* and *minimal rank* of the sum of matrices. Both prove useful in the cryptanalysis of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems.

The main part of this thesis is about concrete trapdoors for the problem of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. We can show that all such systems fall in one of the following four classes: unbalanced oil and vinegar systems (UOV), stepwise triangular systems (STS), Matsumoto-Imai Scheme A (MIA), and hidden field equations (HFE). Moreover, we demonstrate the use of several modifiers. In order to evaluate the security of these four basic trapdoors and their modifiers, we review some cryptanalytic results. In particular, we were able to develop our own contributions in this field by demonstrating an affine approximation attack and an attack using Gröbner base computations against the UOV class. Moreover, we derived a key recovery and inversion attack against the STS class. Using our knowledge of the HFE class, we develop two secure versions of the signature scheme Quartz.

Another important part of this thesis is the study of the key space of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. Using special classes of affine transfor-

mations, denoted "sustainers", we are able to show that all four basic classes have some redundancy in their key spaces and hence, have a smaller key space than previously expected. In particular for the UOV and the STS class, this reduction proves quite dramatic. For HFE and MIA, we only find some minor redundancies. Moreover, we are able to show that our results for MIA are the only ones possible, *i.e.*, there are no other redundancies than the one we describe in this thesis. In addition, we extend our results to several important variations of HFE and MIA, namely HFE-, HFEv, HFEv-, and MIA-. They have been used in practice for the construction of signature schemes, namely Quartz and Sflash.

In order to demonstrate the practical relevance of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic constructions and also of our taxonomy, we show some concrete examples. In particular, we consider the NESSIE submissions Flash, Sflash, and Quartz and discuss their advantages and disadvantages. Moreover, we describe some more recent developments, namely the STS-based schemes enhanced TTS, Tractable Rational Maps, and Rainbow. Then we move on to some application domains for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. In particular, we see applications in the area of product activation keys, electronic stamps and fast one-way functions. Finally, we suggest some new schemes. In particular, we give a generalisation of MIA to odd characteristics and also investigate some other trapdoors like STS and UOV with the branching and the homogenisation modifiers.

All in all, we believe that $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomial systems are a very practical solution to the problem of public key cryptography. At present, it is not possible to use them for encryption. However, we are confident that it will be possible to overcome this problem soon and use $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic constructions both for encrypting and signing.

Samenvatting

Deze thesis geeft een overzicht van het gebruik van stelsels van multivariate kwadratische veeltermen in publieke sleutelcryptografie. In het eerste hoofdstuk worden enkele cryptografische begrippen geïntroduceerd. In het bijzonder wordt de noodzaak van publieke sleutelcryptografie en de alternatieve oplossing ervoor, d.z.w., systemen die noch factorisatie (zoals RSA, Rivest-Shamir-Adleman) noch het discrete logaritme gebruiken (zoals ECC, elliptische krommecryptografie).

Dit wordt gevolgd door een korte inleiding van eindige lichamen en een algemene bespreking over systemen van stelsels van multivariate kwadratische veeltermen en manieren om ze voortestellen. In deze context worden ook affiene transformaties en hun voorstelling besproken. Nadat deze hulpmiddelen zijn geïntroduceerd, worden zij gebruikt om aan te tonen hoe de stelsels van multivariate kwadratische veeltermen voor handtekening- en encryptietoepassingen kunnen worden gebruikt. Bovendien wordt het probleem van uit kwadratische veeltermen bestaande stelsels van vergelijkingen in perspectief gezet en er wordt een verbinding gelegd met de theorie van $\mathcal{NP}$-Volledigheid. Het tweede hoofdstuk besluit met een discussie over de twee verwante problemen *isomorfismen van veeltermen* en *minimale rang* van de som van matrices. Beide blijken nuttig te zijn voor de cryptanalyse van systemen gebaseerd op stelsels van multivariate kwadratische veeltermen.

Het belangrijkste deel van deze thesis gaat over concrete valkuilen voor het probleem van publieke sleutelsystemen die gebaseerd zijn op stelsels van multivariate kwadratische veeltermen. We kunnen aantonen dat al dergelijke systemen in één van de volgende vier klassen vallen: uit evenwicht gebrachte olie-en azijn-systemen (UOV), trapsgewijze driehoekige systemen (STS), Matsumoto-Imai schema A (MIA), en verborgen lichaamsverglijkingen (HFE). Voorts tonen we het gebruik van verscheidene wijzigingen aan. Om de veiligheid van deze vier basis-valkuilen en hun wijzigingen te evalueren, herzien we sommige cryptanalytische resultaten. In het bijzonder kunnen we onze eigen bijdragen op dit gebied ontwikkelen door twee aanvallen tegen UOV aan te tonen, namelijk een benaderingsaanval en een aanval die Gröbnerbasis berekeningen gebruikt. Voorts leidden

we een sleutelterug-win en een inversie-aanval tegen de klasse STS af. Gebruik makend van onze kennis over HFE, ontwikkeln we twee veilige versies van het Quartz handtekeningsschema.

Een ander belangrijk deel van deze thesis is de studie van de sleutelruimte van publieke sleutelcryptografie gebaseerd op stelsels van multivariate kwadratische veeltermen. Gebruikmakend van speciale klassen van affiene transformaties, die we "onderhouders" noemen, kunnen we aantonen dat alle vier de basisklassen wat redundantie in hun sleutelruimte hebben en dus een kleinere sleutelruimte bevatten dan eerder verwacht. In het bijzonder blijkt deze reductie voor UOV en de STS klasse vrij dramatisch. Voor HFE en MIA vinden we slechts enkele minder belangrijke reduntanties. Voorts kunnen we aantonen dat onze resultaten voor MIA de enige mogelijke zijn, d.z.w., zijn er geen andere redundanties dan degene die we in deze thesis beschrijven. Bovendien breiden we onze resultaten tot verscheidene belangrijke variaties van HFE en MIA uit, namelijk HFE-, HFEv, HFEv-, en MIA-. Deze zijn in de praktijk gebruikt voor de bouw van handtekeningsschema's, namelijk Quartz, Flash en Sflash.

Om de praktische relevantie van de bouw van systemen gebaseerd op stelsels van multivariate kwadratische veeltermen en ook van onze classificatie aan te tonen, geven we enkele concrete voorbeelden. In het bijzonder behandelen we de NESSIE inzendingen, Flash, Sflash en Quartz en bespreken we hun voor- en nadelen. Voorts beschrijven we wat meer recente ontwikkelingen, namelijk op STS-gebaseerde "verbeterde TTS", "Handelbare Rationele Transformaties", en Regenboog. Vervolgens begeven we ons naar de toepassingsgebieden van publieke sleutelcryptografie systemen gebaseerd op stelsels van multivariate kwadratische veeltermen. In het bijzonder behandelen we toepassingen op het gebied van product-activeringssleutels, elektronische postzegels en snelle eenrichtingsfuncties. Ten slotte stellen we enkele nieuwe schema's voor. In het bijzonder geven we een generalisatie van MIA voor oneven karakteristiek, en onderzoeken we ook andere valkuilen zoals STS en UOV met vertakking- en homogenisatieaanpassers. Al met al, geloven we dat de uit stelsels van multivariate kwadratische veeltermen bestaande systemen een zeer praktische oplossing geven voor het probleem van publieke sleutelcryptografie. Momenteel is het niet mogelijk om ze voor encryptie te gebruiken. Nochtans zijn we zeker dat het mogelijk zal zijn om dit probleem te overwinnen, en om de stelsels van multivariate kwadratische veeltermen te gebruiken voor zowel encryptie als voor het zetten van handtekeningen.

Zusammenfassung

Diese Dissertation gibt einen Überblick über $\mathcal{M}$ultivariate $\mathcal{Q}$uadratische Polynomgleichungen und ihre Verwendung in der asymmetrischen Kryptographie.

Das erste Kapitel führt in einige allgemeine Begriffe der Kryptographie ein. Insbesondere wird die Notwendigkeit von Public-Key Kryptographie sowie alternativer Schemata motiviert. Unter alternativen Schemata verstehen wir in diesem Kontext Systeme, die weder Faktorisierung benutzen (wie z.B. RSA, Rivest-Shamir-Adleman) noch diskrete Logarithmen (wie z.B. ECC, Elliptic Curve Cryptographie).

Darauf folgt eine kurze Einführung in endliche Körper sowie eine allgemeine Diskussion über $\mathcal{M}$ultivariate $\mathcal{Q}$uadratische Gleichungssysteme und ihre Darstellungsformen. In diesem Kontext behandeln wir affine Transformationen und deren Representation. Nachdem diese Werkzeuge eingeführt wurden, benutzten wir sie um zu zeigen, wie $\mathcal{M}$ultivariate $\mathcal{Q}$uadratische Systeme für elektronische Unterschriften und Verschlüsselung verwendet werden können. Des weiteren zeigen wir eine Verbindung zwischen $\mathcal{M}$ultivariaten $\mathcal{Q}$uadratischen Systemen und der Theorie der $\mathcal{NP}$-Vollständigkeit auf. Das zweite Kapitel endet mit den beiden verwandten Problemen *Isomorphismus von Polynomen* und *Minimaler Rang* der Summe von Matrizen. Beide sind für die Kryptanalyse von $\mathcal{M}$ultivariaten $\mathcal{Q}$uadratischen Systemen von hohem Wert.

Der Hauptbeitrag dieser Dissertation sind konkrete Falltüren für die Konstruktion $\mathcal{M}$ultivariater $\mathcal{Q}$uadratischer Systeme. Wir können zeigen, dass alle diese Systeme in eine der folgenden vier Klassen fallen: Unbalanced Oil and Vinegar Systeme (UOV), Stepwise Triangular Systeme (STS), Matsumoto-Imai Schema A, und Hidden Field Equations (HFE). Des weiteren zeigen wir die Verwendung mehrerer Modifizierer. Um die Sicherheit dieser vier grundlegenden Falltüren sowie der Modifizierer einschätzen zu können, besprechen wir einige kryptanalytischen Ergebnisse. Unsere eigenen Beiträge auf diesem Gebiet sind ein affiner Approximierungsangriff sowie ein Angriff, der auf der Berechnung von Gröbner Basen beruht. Beide Angriffe erfolgten gegen UOV. Des weiteren haben wir sowohl einen Key-Recovery-Angriff wie auch einen Inversionsangriff gegen

STS entwickelt. Unter Ausnutzung unserer Kenntnis der HFE-Klasse konnten wir zwei sichere Versionen des Signatursystems Quartz entwickeln.

Ein weiterer wichtiger Teil dieser Dissertation ist das Studium des Schlüsselraums $\mathcal{M}$ultivariater $\mathcal{Q}$uadratischer Systeme. Durch Benutzung spezieller Klassen affiner Transformationen, sog. "Erhalter", können wir zeigen, dass alle vier Grundklassen einen redundanten Schlüsselraum haben. Der Schlüsselraum ist damit kleiner als ursprünglich angenommen. Vor allem für UOV und STS ergibt sich eine sehr starke Reduktion, während wir für HFE und MIA nur kleinere Redundanzen finden konnten. Des weiteren waren wir im Stande zu zeigen dass die von uns gefundenen Redundanzen in der MIA-Klasse die einzig möglichen sind. Wir konnten unsere Ergebnisse zu wichtigen Varianten von HFE und MIA, nämlich HFE-, HFEv, HFEv-, und MIA- verallgemeinern. Alle vier wurden in der Praxis für die Konstruktion von Unterschriftenschemata benutzt, nämlich Quarz und Sflash.

Um die praktische Bedeutung sowohl $\mathcal{M}$ultivariater $\mathcal{Q}$uadratischer Systeme wie auch unserer Taxonomie zu demonstrieren, diskutieren wir einige konkrete Beispiele. Wir konzentrieren uns dazu vor allem auf die NESSIE-Einreichungen Flash, Sflash und Quartz und besprechen ihre Vor- und Nachteile. Des weiteren beschreiben wir einige neuere Entwicklungen in Gestalt der STS-basierten Schemata enhanced TTS, Rational Tractable Maps, und Rainbow. Darauf folgen einige mögliche Anwendungsfelder $\mathcal{M}$ultivariater $\mathcal{Q}$uadratischer Systeme wie Produktaktivierungsschlüssel, elektronische Briefmarken und schnelle Einwegfunktionen. Letztendlich schlagen wir einige neue Schemata vor. Insbesondere verallgemeinern wir MIA auf ungerade Charakteristiken und untersuchen auch andere Falltüren wie STS und UOV mit dem Homogenisierungs- und dem Verzweigungsmodifizierer.

Alles in allem glauben wir, dass $\mathcal{M}$ultivariate $\mathcal{Q}$uadratische Systeme eine sehr praktische Lösung für das Primitiv „asymmetrisch Kryptographie“ darstellen. Zur Zeit ist es leider nicht möglich, sie für Verschlüsselungsschemata zu verwenden. Wir sind allerdings zuversichtlich, dass sich dieses Problem bald überwinden lässt und $\mathcal{M}$ultivariate $\mathcal{Q}$uadratische Systeme damit sowohl für Verschlüsselung wie auch für elektronische Unterschriften eingesetzt werden können.

Contents

List of Figures

List of Tables

List of Symbols

Basic integers

a	number of added equations (+ modification)
B	branching number ($\perp$ modification)
f	number of fixed variables (f modification)
h	number of homogenizing variables (h modification)
L	levels (STS class)
o	number of oil variables (UOV class)
q	number of elements in the ground field
r	step-width (STS class)
	number of equations removed (- modification)
v	number of vinegar variables (UOV class)
	number of vinegar variables (v modification)
w	perturbation dimension (i modification)

Basic sets

$\mathbb{F}$, GF(q)	finite field with q elements
$\mathbb{N}$	positive integers, *i.e.*, the set $\{1, 2, 3, \ldots\}$
$\mathbb{N}_0$	non-negative integers, *i.e.*, the set $\mathbb{N} \cup \{0\}$
$\mathbb{R}$	real numbers
$\mathbb{Z}$	integers, *i.e.*, the set $\{a, -a : a \in \mathbb{N}_0\}$
$\mathbb{Z}^+$	positive integers, *i.e.*, $\mathbb{N}$

Derived sets

$\mathrm{Aff}_0(\mathbb{F}^n, \mathbb{F}^m)$	all affine transformations $\mathbb{F}^n \to \mathbb{F}^m$
$\mathrm{Aff}_0(\mathbb{F}^n)$	affine transformation over $\mathbb{F}^n$, *i.e.*, the set $\mathrm{Aff}_0(\mathbb{F}^n, \mathbb{F}^n)$
$\mathrm{Aff}^{-1}(\mathbb{F}^n, \mathbb{F}^m)$	surjective affine transformations $\mathbb{F}^n \to \mathbb{F}^m$
$\mathrm{Aff}^{-1}(\mathbb{F}^n)$	invertible affine transformations $\mathbb{F}^n \to \mathbb{F}^n$, *i.e.*, the set $\mathrm{Aff}^{-1}(\mathbb{F}^n, \mathbb{F}^n)$
$\mathbb{E}$	finite field of dimension n over $\mathbb{F}$ with q^n elements
$\mathbb{E}^*$	$\mathbb{E}\backslash\{0\}$, *i.e.*, the multiplicative group of $\mathbb{E}$
$\mathbb{F}^*$	$\mathbb{F}\backslash\{0\}$, *i.e.*, the multiplicative group of $\mathbb{F}$
$\mathbb{F}^n$	n-dimensional vectorspace over $\mathbb{F}$
$\mathbb{F}^{n\times m}$	set of $(n \times m)$ matrices over $\mathbb{F}$
$\mathrm{Hom}_0(\mathbb{F}^n, \mathbb{F}^m)$	all homomorphic transformations $\mathbb{F}^n \to \mathbb{F}^m$
$\mathrm{Hom}_0(\mathbb{F}^n)$	homomorphic transformation over $\mathbb{F}^n$, *i.e.*, the set $\mathrm{Hom}_0(\mathbb{F}^n, \mathbb{F}^n)$
$\mathrm{Hom}^{-1}(\mathbb{F}^n, \mathbb{F}^m)$	surjective homomorphic transformations $\mathbb{F}^n \to \mathbb{F}^m$
$\mathrm{Hom}^{-1}(\mathbb{F}^n)$	invertible homomorphic transformations $\mathbb{F}^n \to \mathbb{F}^n$, *i.e.*, the set $\mathrm{Hom}^{-1}(\mathbb{F}^n, \mathbb{F}^n)$
$\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$	$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials in n input variables and m output variables
$\mathcal{MQ}(\mathbb{F}^n)$	$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials in n input variables and n output variables, *i.e.*, $\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^n)$
$\mathcal{MQ}$-D	$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic problem over the Domain D
$\mathcal{MQ}$-GF(2)	$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic problem over GF(2)
S_n	set of all permutations of $\{1, \ldots, n\}$

Operators

^{t}	matrix or vector transposition
$a \,\|\|\, b$	concatenation of the two strings a and b
$\lfloor a \rfloor$	floor of $a \in \mathbb{R}$, *i.e.*, the integer $\max\{b \in \mathbb{Z} : a \geq b\}$
$\lceil a \rceil$	ceiling of $a \in \mathbb{R}$, *i.e.*, the integer $\min\{b \in \mathbb{Z} : a \leq b\}$
$\mathrm{char}\ \mathbb{F}$	characteristic of the field $\mathbb{F}$
$Diag(x_1, \ldots, x_n)$	diagonal matrix in $\mathbb{F}^{n\times n}$ with diagonal coefficients $x_1, \ldots, x_n \in \mathbb{F}$

List of Abbreviations

cf	*confer*, compare
CPC	Chained Patarin Construction
enTTS	enhanced Tame Transformation Signatures
FPC	Feistel-Patarin Construction
gSTS	general STS
HFE	Hidden Field Equations
IP	Isomorphism of Polynomials
MIA	Matsumoto-Imai Scheme A
$\mathcal{MQ}$	$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic
$\mathcal{NP}$	Non-deterministic Polynomial
OV	Oil and Vinegar
RSA	Rivest-Shamir-Adleman
RSE(2)PKC	Random Simultaneous Equations Degree 2 Public Key Cryptosystem
RSSE(2)PKC	Random Singular Simultaneous Equations Degree 2 Public Key Cryptosystem
rSTS	regular STS
SME	Simultaneous Multivariate Equations
SSL	Secure Sockets Layer
STS	Stepwise Triangular Systems
TLS	Transport Layer Security
TRMS	Tractable Rational Map Signatures
TTS	Tame-Transformation Signatures
UOV	Unbalanced Oil and Vinegar
w.l.o.g.	without loss of generality

Chapter 1

Introduction

In today's Internet driven society, information is a valuable asset: it is "produced", stored, and traded. So all this information needs protection of several kinds: protection against altering, certainty of origin, and also protection of its very content from leaking out or being stolen. For some of these aims, mechanisms we know from the physical world are sufficient: for example, the access to a computer class is usually strictly guarded: either we have a human supervisor who makes sure that only legitimate people may enter this computer class, or we have some kind of physical access control: either by keys or key cards. However, as soon as information is exchanged over untrusted networks such at the open Internet, this physical protection becomes vain: for example, it is impossible to protect the whole Internet this way as it belongs to too many people living in too many countries. Hence, we need a different toolkit for ensuring authenticity and secrecy of the information flowing through open networks as the Internet. One of the most prominent examples for such a toolkit is called "cryptology".

1.1 Cryptology

Although cryptography received very much attention during the last decades, the science of "secret writing" and its analysis are actually far older. In this section we give a brief overview of some important steps in the development of cryptology from its early beginning to nowadays "public key" cryptology. This way, we motivate the research carried out in this doctorate.

1.1.1 The Caesar Cipher

One early example of cryptology can be found in the times of the Roman empire. It is a cipher called "Caesar" (after Julius Caesar) and was used both by him and his successors [Bau95, Sec. 3.2.3]. The following example illustrates this technique:

clear text	a	t	t	a	c	k	a	t	d	a	w	n
cipher text	**D**	**W**	**W**	**D**	**F**	**N**	**D**	**W**	**G**	**D**	**N**	**Q**

Here the clear text "attack at dawn" is transformed into the corresponding cipher text "**DWWDFNDWGDNQ**". For a Caesar cipher, each letter is replaced by its third successor in the alphabet, *i.e.*,

$$\begin{array}{ccccccccc} \text{a} & \rightarrow & \textbf{D,} & \text{b} & \rightarrow & \textbf{E,} & \text{c} & \rightarrow & \textbf{F,} \\ & & & & \cdots & & & & \\ \text{x} & \rightarrow & \textbf{A,} & \text{y} & \rightarrow & \textbf{B,} & \text{z} & \rightarrow & \textbf{C} \end{array}$$

To decrypt a message which has been sent in Caesar, each letter has to be shifted back 3 positions in the alphabet. Variations of the Caesar cipher use not only a simple shift by 3 but any permutation of the alphabet, *i.e.*, they use a more complex key. However, due to their rather simple structure, these kind of ciphers can easily be broken using the frequency of different letters in different languages. For example, the letter "e" is the most frequent one in English (12.51%) and also in German (17.48%) [Bau95, Fig. 91]. But the shorter the text, the lesser the likelihood that the text shows the same frequency as the whole language. For example, "attack at dawn" contains only the vowel "a" and no "e" at all. Still, the Caesar cipher has to be considered broken today, even for such short texts.

1.1.2 One-Time Pad

A logical extension of the substitution principle used in the Caesar cipher is to have not one but many different substitutions. The extreme case — one independent substitution for each letter — is called a "one-time pad" [Sch96, Sec. 1.5]. For a one-time pad, the key has to be as long as the message to be sent and also "truly random". In addition, this key can never be used again (therefore the name "one-time pad"). If it is reused, this is a very serious breach of security. However, in case all the above criteria are met, a one-time-pad is a provably perfect secure system, *i.e.*, as long as the key is kept secret, no eavesdropper can reveal the message [Beu93, Sec. 3.3]. The technique is illustrated in the following example, cf Table 1.1.

Here the secret key chooses the substitution to be used, *i.e.*, how many letters a character is to be shifted. If the secret key is an "x" (as for the first "a" in

Table 1.1: Example of the One-Time Pad

clear text	a	t	t	a	c	k	a	t	d	a	w	n
secret key	*x*	*v*	*d*	*y*	*w*	*b*	*z*	*f*	*o*	*h*	*c*	*e*
cipher text	**Y**	**P**	**X**	**Z**	**Z**	**M**	**A**	**Z**	**S**	**I**	**Z**	**S**

"attack"), the corresponding clear text character is shifted by 24 positions. As the alphabet has 26 characters, a shift by 26 positions (*i.e.*, for the secret key character "*z*"), does not have any effect, so the letter stays unchanged. This can be seen for the "a" in "at". To decrypt, each step is reversed, *i.e.*, instead of using the i^{th} successor of a letter, we use its i^{th} predecessor — with i depending on the key-letter.

Intuitively, the one-time pad is secure as all messages are equally alike. In the above example, we can change "attack at dawn" to "surrender now" by "just" changing the secret key, cf Table 1.2. Hence, without knowing the secret key, the attacker cannot "learn" anything new as the cipher does not reveal anything about the distribution of the secret key. So as long as each key is used only one, we have an unbreakable cipher.

Table 1.2: Second Example of the One-Time Pad

clear text	s	u	r	r	e	n	d	e	r	n	o	w
secret key	*f*	*u*	*f*	*h*	*u*	*y*	*w*	*u*	*a*	*u*	*k*	*v*
cipher text	**Y**	**P**	**X**	**Z**	**Z**	**M**	**A**	**Z**	**S**	**I**	**Z**	**S**

As there is a perfectly secure cipher available — in fact, this cipher was invented nearly a century ago in 1917 by Joseph Mauborgne and Gilbert S. Vernam [Sch96, Sec. 1.5] — one would expect cryptology to have come to an end as there seems to be no need for any other ciphers. But the contrary is true as there is a high price to pay for perfect security: the key has to be as long as the message and also perfectly random. The latter requires a "random source", *e.g.*, radioactive decay or thermal noise [Sch96, Sec. 17.14]. In addition, it needs very much randomness, as the key has to be very large. For example, to encrypt an ISDN connection (64,000 bits/second) for only 1 minute, we need approx. 0.5 MBytes — which is the equivalent of a book with approx. 200 pages. For high bandwidth channels like Ethernet (10-100 Mbits/second), the required key length is far beyond any reasonable size — even the equivalent of a multi-volume lexicon would not last for long. Moreover, the sender and the receiver of a message need to know the same key, therefore, it has to be transmitted securely to at least one of them. If there are many keys and if they are also rather large, this can be a challenging task. In addition, both sender and recipient have to store the key in a secure manner. So in practice, a one-time pad can only be used if very few messages are transmitted and if they are also rather short. Alternatively, if the security requirements are very high. For example, the red telephone between the White House and the Kremlin used to be protected by a one-time pad [Mas92, p. 11]. According to [Mas92, p. 11], they use a symmetric cipher, at least since the early 90s. Hence we see that perfect security is not always necessary in practice — even on the highest political level.

1.1.3 Modern Cryptology

Nowadays, cryptology does not only deal with the problem of securing communication against eavesdropping, but also with problems such as message integrity and authentication (see above).

For a data authentication problem, the attacker plays a more active role than only eavesdropping, *i.e.*, she (in cryptographic papers, the attacker is often called "Eve") does not only try to eavesdrop but also to alter the content of a message. This can be very dangerous if, *e.g.*, the sum on a money transfer is changed: instead of withdrawing 100 euro from a customer's bank account, the bank takes 1,000,000 — assuming that the attacker altered the content of the money transfer form on the way from the customer to the bank.

In this context, a comparably new development in cryptography, the so-called "public key cryptography" is very useful. It dates back to 1976 when Diffie and Hellman published their paper *"New Directions in Cryptography"* [DH76]. They introduced the idea of not using one single, secret key for both encryption and decryption (as we saw it in the previous examples) but one key for encryption (called "public key") and one key for decryption (called "private key"). This looks rather complicated but has nice advantages over the old, so-called "secret key cryptography": users no longer have to care about many secret keys (one for each communication partner), but only one, *i.e.*, their own private key. The other keys can (and in fact, should) be publicly known, so there is no need to keep them away from eavesdroppers. Moreover, applications such as digital signatures of electronic documents become possible now. For electronic signatures, the private key is used to generate a signature ("to sign a document"), while the public key checks this signature ("to verify a signature"). We see an example of this signing technique in Section 2.3.

Although it has many practical advantages, at first the idea of Diffie and Hellman was purely theoretical, *i.e.*, they did not present an algorithm which could be used for public key cryptography but stated the mere principle. However, during the following years, such algorithms were developed, *e.g.*, RSA or ElGamal. The first is based on the problem of factoring large numbers (1024 bits and more), the latter on a problem called "discrete logarithm", *i.e.*, to compute the solution of equations like $a^x = b$ for given a, b in discrete equations. Both are computationally difficult problems — even with modern algorithms and computers. See [MvOV96] for these and other cryptographic algorithms.

More formally, we have the following construction in public key cryptography: let k_A be Alice's private key and K_A her public key. For a public key encryption scheme, we need two functions $c := E(K_A, m)$ and $m' := D(k_A, c)$ such that $m' = m \ \forall m$. Moreover, for given public key K_A it must be easy to compute $E(K_A, m)$ for any given message m. Still, the computation of any function $m'' := D(c)$ without the knowledge of the private key k_A or some equivalent information may not be possible. Similar, there may not be an efficient function $k'_A := f(K_B)$ which computes an equivalent version of Alice's private key. For a signature scheme, we have a similar notion: here we have the two functions $s := S(k_A, m)$ and $b := V(K_B, s, m)$, namely signature computation $S(\cdot, \cdot)$ and signature verification $V(\cdot, \cdot, \cdot)$. As for public key encryption schemes, it is vital that it is not possible to compute the private key efficiently from the public key, nor to derive signatures without this private key. As we will see in the sequel, these requirements are met by most schemes discussed in this thesis.

Nowadays, cryptographic techniques are widely used — both by civil and military users. In non-military use, e-commerce is an important application domain of cryptographic techniques. For example, if you buy a book at Amazon, your credit card information is secured through the SSL/TLS protocol which makes use of certificates, *i.e.*, electronic signatures on public keys, public key exchange routines to transmit so-called "session keys", and also symmetric key algorithms, which use the session key to encrypt the information you transmit to Amazon and vice versa [DA99, Shi00].

1.2 Motivation

As we saw in the previous section, public key cryptography is an important tool for nowadays information society. Unfortunately, the security of public key schemes used in practice relies on a rather small number of problems: either factoring (RSA) or discrete logarithms (ECC). Both problems are *currently* considered to be hard. It is widely believed that research on new schemes based on other classes of problems is necessary. Such work provides greater diversity and hence forces cryptanalysts to spend additional effort concentrating on completely new types of problems. This way, we make sure that not all "crypto-eggs" are in one basket. To strengthen the necessity for new schemes, we want to point out that important results on the potential weaknesses of existing public key schemes are emerging. In particular techniques for factorisation and solving discrete logarithm improve continually. For example, polynomial time quantum algorithms [Sho97] can be used to solve both problems. Therefore, the existence of quantum computers in the range of 1000 bits would be a real-world threat to systems based on factoring or the discrete logarithm problem. This stresses the importance of research into new algorithms for asymmetric cryptography.

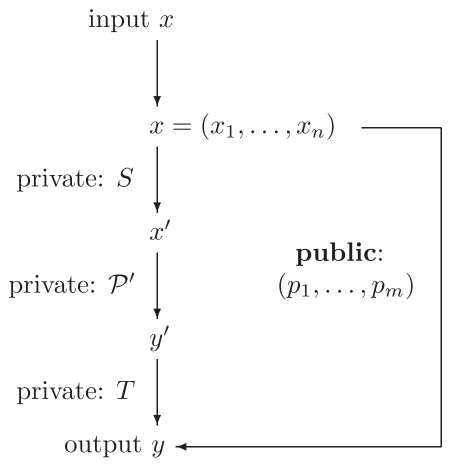

Figure 1.1: Outline of the General $\mathcal{MQ}$-trapdoor

One proposal for secure public key schemes is based on the problem of solving $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations ($\mathcal{MQ}$-problem) over finite fields. All of these proposals share the same type of public key, *i.e.*, polynomials of degree 2 over (small) finite fields, cf Figure 1.2 for an overview and Section 2 for a more formal treatment. In Figure 1.2 we have $1 \leq i \leq m$ polynomials in $1 \leq j \leq k \leq n$ variables each, and the coefficients $\gamma_{i,j,k}, \beta_{i,j}, \alpha_i \in \mathbb{F}$ where $\mathbb{F}$ denotes some finite field. The first such proposal is due to Matsumoto and Imai [IM85]. In the last 15 years, several such public key cryptoschemes (PKC) have been proposed. A typical multivariate public key system uses a private key $T \circ \mathcal{P}' \circ S$ where $\circ$ denotes the composition of functions, cf Figure 1.1. Here, $S \in \text{Aff}^{-1}(\mathbb{F}^n)$ and $T \in \text{Aff}^{-1}(\mathbb{F}^m)$ represent two affine transformations over the finite field $\mathbb{F}$. The central map $\mathcal{P}'$ consists of m central equations in n variables each. For an $\mathcal{MQ}$-scheme, the degree of these equations is 2. Moreover, the central map $\mathcal{P}'$ must be easy to invert to allow the decryption or signing of messages. So the secret key of the $\mathcal{MQ}$-system is composed of the triple $(S, \mathcal{P}', T) \in \text{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \text{Aff}^{-1}(\mathbb{F}^m)$. We want to point out that the different proposals only differ in the

$$\begin{array}{rcl}
p_1(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{1,j,k}x_jx_k + \sum_{j=1}^{n}\beta_{1,j}x_j + \alpha_1 \\
& \vdots & \\
p_i(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{i,j,k}x_jx_k + \sum_{j=1}^{n}\beta_{i,j}x_j + \alpha_i \\
& \vdots & \\
p_m(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{m,j,k}x_jx_k + \sum_{j=1}^{n}\beta_{m,j}x_j + \alpha_m
\end{array}$$

Figure 1.2: Public Key for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Public Key Systems

structure of their central equations $\mathcal{P}'$. Hence, depending on this structure, we are able to identify several classes: the MIA schemes [IM85, MI88], HFE-like schemes [Pat96b, Cou01], Unbalanced Oil and Vinegar schemes [KPG99], and Stepwise Triangular Systems [WBP04]. All of them rely on the fact that the $\mathcal{MQ}$-problem, *i.e.*, finding a solution $x \in \mathbb{F}^n$ for a given system $\mathcal{P}$ is computationally difficult, namely $\mathcal{NP}$-complete, cf Section 2.5. In addition, factoring $\mathcal{P}$ into its components $T, \mathcal{P}', S$ is considered to be a hard problem if $S, \mathcal{P}', T$ do not have a special structure. This problem has previously been studied under the name Isomorphism of Polynomials Problem, cf Section 2.6.1.

1.3 Related Work

As outlined above, we concentrate on $\mathcal{M}$ultivariate $\mathcal{Q}$*uadratic* equations over finite fields in this text. They have the nice property that an attacker does not even know which type of scheme he attacks, given the public key alone, *i.e.*, we have a kind of "secret public key schemes" [Pat00]. When deciding on the scope of this thesis, we had to draw the line somewhere. As an objective criterion, we used the degree of the public key polynomials, *i.e.*, other multivariate schemes which are based on equations of higher degree are not considered. This includes in particular the polynomial substitution scheme of [FD85] and the Dragon scheme from [Pat96a] as they have a public key of higher degree. In this context we also want to mention the Bi-Quadratic C* scheme of [DF05]. Similarly, we do not consider birational permutations [Sha93], as they are based on finite rings rather than finite fields. Moreover, they have been successfully cryptanalysed in [CSV93, The95, CSV97]. In addition, we do not consider the matrix based schemes from [PGC98a] either, as its use has been strongly discouraged in the very paper were it has been developed, and also is not an $\mathcal{MQ}$-system in this stronger sense.

In this context we refer to Section 2.2: there, we outline while quadratic equations play such a prominent role. In a nutshell, they are already $\mathcal{NP}$-complete while the number of coefficients is still reasonably small. This also stresses why we decided to concentrate on schemes based on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations rather than including schemes of higher degree.

1.4 Achievement

This thesis contains several new results. First, it gives an overview on the whole area of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic cryptography. Until now, it was necessary to read several papers — each of them using their own notation — to get a good idea on this area. In this context, we also established that there are only 4 basic trapdoors and 10 generic modifiers. Before, this was not obvious at all. More importantly and to derive this taxonomy, we needed to generalise existing trapdoors, *i.e.*, the TPM trapdoor of [GC00] was generalised to STS [WBP04], and the MIA trapdoor was generalised to odd characteristics (MIO, cf Section 6.4.1). Similar, the modifiers sparse polynomials ("s"), vinegar variables ("v"), and internal perturbation ("i") have not been treated in their full generality before. In addition, the homogenising ("h") modifier has been developed during this thesis. Quite interesting from a mathematical point of view is the question of equivalent keys of $\mathcal{MQ}$-schemes. Surprisingly, this was not addressed in the open literature before. The first publication we are aware of in this context is [WP05c]. The theory described there has been further developed in this thesis and can be found in Section 5. Apart from this mathematical work, we also successfully cryptanalysed the UOV class [BWP05] and the STS class [WBP04]. Moreover, we developed a variation of the proposed standard "Quartz" which is secure against all known attacks [WP04]. Using the knowledge developed in this thesis, we investigated the question of possible applications of $\mathcal{MQ}$-schemes. In a nutshell, we see applications in the context of electronic stamps, product activation keys, and fast one-way functions [WP05a].

1.5 Outline

This thesis is organised as follows: following this introduction we consider some basic mathematical tools, necessary for the understanding of $\mathcal{MQ}$-schemes. After this, we consider some basic constructions for embedding trapdoors into $\mathcal{MQ}$-systems. This is followed by some cryptanalytic results and the question of equivalent keys. The last two chapters deal with possible applications and conclusions.

Chapter 2

The General $\mathcal{MQ}$-construction

After motivating the topic of this thesis, *i.e.*, $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems, we now move on by introducing some properties and notations useful for the remainder of this thesis. In particular, we will develop the mathematical tools necessary for understanding $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. We start with briefly introducing finite fields, then concentrating on the general problem of multivariate polynomial equations, an alternative matrix representation, consider affine transformations, and also the two related problems MinRank and Isomorphism of Polynomials.

Our own achievement in this chapter are easier proofs for the lemmata and theorems stated here. Moreover, we simplified and unified the notation used in the context of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems.

2.1 Finite Fields

As finite fields are a very basic building block for these kind of schemes, we start with properly introducing them. Loosely speaking, a (finite) field consists of a (finite) set of elements, and two operations, namely addition (denoted "+") and multiplication (denoted "·"). These operations need to fulfil certain criteria:

DEFINITION 2.1.1 *Let* $\mathbb{F}$ *be a set of* $q \in \mathbb{N}$ *elements with the two operations addition* $+ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ *and multiplication* $\cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$. *Note that this definition implies closure for addition and multiplication. We call* $(\mathbb{F}, +, \cdot)$ *a* field *if the following axioms are fulfilled:*

1. additive Abelian group $(\mathbb{F}, +)$:
   (a) associativity: $\forall a, b, c \in \mathbb{F} : ((a + b) + c) = (a + (b + c))$
   (b) additive neutral: $\exists e \in \mathbb{F} : \forall a \in \mathbb{F} : a + e = a$.
   In the remainder of this thesis, we denote this e with 0
   (c) additive inverse: $\forall a \in \mathbb{F}\ \exists a' \in \mathbb{F} : a + a' = 0$.
   In the remainder of this thesis, we denote this a' with $-a$
   (d) commutativity: $\forall a, b \in \mathbb{F} : a + b = b + a$
2. multiplicative Abelian group $(\mathbb{F}^*, \cdot)$ *for* $\mathbb{F}^* := \mathbb{F} \setminus \{0\}$:
   (a) associativity: $\forall a, b, c \in \mathbb{F} : ((a \cdot b) \cdot c) = (a \cdot (b \cdot c))$
   (b) multiplicative neutral: $\exists e \in \mathbb{F} : \forall a \in \mathbb{F} : a \cdot e = a$.
   In the remainder of this thesis, we denote this e with 1
   (c) multiplicative inverse: $\forall a \in \mathbb{F}^*\ \exists a' \in \mathbb{F}^* : a \cdot a' = 1$.
   In the remainder of this thesis, we denote this a' with a^{-1}
   (d) commutativity: $\forall a, b \in \mathbb{F} : a \cdot b = b \cdot a$
3. distributivity: $\forall a, b, c \in \mathbb{F} : a \cdot (b + c) = a \cdot b + a \cdot c$

Remark 2.1.2 *For brevity, we write ab instead of $a \cdot b$. If it is clear from the context which addition and multiplication we use with the field, we also write $\mathbb{F}$ instead of $(\mathbb{F}, +, \cdot)$.*

DEFINITION 2.1.3 *Let q be a prime number, $\mathbb{F} := \{0, \ldots, q-1\}$, and addition and multiplication usual integer addition and multiplication modulo this prime number q. Then we call $(\mathbb{F}, +, \cdot)$ a* prime field.

DEFINITION 2.1.4 *Let $\mathbb{F}$ be a field and $i(t) \in \mathbb{F}[t]$ an irreducible univariate polynomial in the variable t over $\mathbb{F}$ with degree n. Furthermore, we define the set $\mathbb{E} := \mathbb{F}[t]/i(t)$ as equivalence classes of polynomials modulo $i(t)$ and the operation addition "+" as normal addition of polynomials, and "·" multiplication of polynomials modulo the irreducible polynomial $i(t)$. Then we call $(\mathbb{E}, +, \cdot)$ a* polynomial field *and also say that it is a degree n* extension *of the* ground field $\mathbb{F}$.

We want to point out that definitions 2.1.1, 2.1.3, and 2.1.4 are consistent: it it possible to prove that the construction from the two latter comply with the field axioms from the first. Moreover, all finite fields are either of the prime field or the polynomial field type. The corresponding proofs and further properties of finite fields can be found in [LN00]. In particular, we want to stress the following

Lemma 2.1.5 *Let $\mathbb{F}$ be a finite field and let $q := |\mathbb{F}|$ be the number of its elements. Then we have $\forall x \in \mathbb{F} : x^q = x$* (Frobenius automorphism).

This lemma will prove particularly useful in the context of schemes defined over extension fields (cf sections 3.1.3 and 3.1.4) and in the context of affine transformations (cf Section 2.2.3). For efficient implementation of arithmetic on finite fields we refer the reader to [BSS99, LD00].

The last piece we need before moving on to $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials is an isomorphism between the extension field $\mathbb{E}$ of dimension n over the ground field $\mathbb{F}$ (cf Definition 2.1.4) and the vector space $\mathbb{F}^n$. To this aim, we observe that all field elements $a \in \mathbb{E}$ have the form

$$a_{n-1}t^{n-1} + \cdots + a_1 t + a_0 \text{ with } a_i \in \mathbb{F}\,.$$

In addition, we see that a vector $b \in \mathbb{F}^n$ can be represented as $(b_1, \ldots, b_n)$ with $b_i \in \mathbb{F}$.

Definition 2.1.6 *Let $\mathbb{E}$ be an n^{th}degree extension of the ground field $\mathbb{F}$ and $\mathbb{F}^n$ the corresponding vector space. Then we call $\phi : \mathbb{E} \to \mathbb{F}^n$ with*

$$\phi(a) := b \text{ and } b_i := a_{i-1} \text{ for } 1 \le i \le n$$

for $a_0, \ldots, a_{n-1}, b_1, \ldots, b_n \in \mathbb{F}$ as defined above the canonical bijection *between $\mathbb{E}$ and $\mathbb{F}^n$. We also use its inverse ϕ^{-1} and have $\phi(\phi^{-1}(b)) = b$ for all $b \in \mathbb{F}^n$ and $\phi^{-1}(\phi(a)) = a$ for all $a \in \mathbb{E}$.*

2.2 Considerations about Multivariate Polynomial Equations

After introducing finite fields, we move on to the problem of solving a system of multivariate polynomial equations.

2.2.1 General Multivariate Polynomial Equations

Let $n \in \mathbb{N}$ be the number of variables, $m \in \mathbb{N}$ the number of equations, and $d \in \mathbb{N}$ the degree of the system. Here $x_1, \ldots, x_n$ are variables over $\mathbb{F}$. By convention, we set $x_0 := 1$, *i.e.*, the multiplicative neutral element in $\mathbb{F}$. For given $n, d \in \mathbb{N}$ and using a vector v with components $v_1, \ldots, v_d \in \{0, \ldots, n\}$ we define

$$\mathcal{V}_n^d := \begin{cases} \{0\} & \text{for } d = 0 \\ \{v \in \{0, \ldots, n\}^d : i \le j \Rightarrow v_i \le v_j\} & \textit{otherwise} \end{cases}$$

We are now able to state the problem of multivariate polynomial equations. Let $\mathcal{P}$ be a system of m polynomials in n variables with maximum degree $d \in \mathbb{N}$ each, *i.e.*, we have $\mathcal{P} := (p_1, \ldots, p_m)$ where all p_i have the form

$$p_i(x_1, \ldots, x_n) := \sum_{v \in \mathcal{V}_n^d} \gamma_{i,v} \prod_{j=1}^{d} x_{v_j} \text{ for } 1 \leq i \leq m$$

with the coefficients $\gamma_{i,v} \in \mathbb{F}$ and vectors $v \in \mathcal{V}_n^d$.

This allows us to define the problem of Simultaneous Multivariate Equations (SME): Let $y_1, \ldots, y_m \in \mathbb{F}$ be some field elements and multivariate polynomials $p_1, \ldots, p_m$ defined as above. Then finding a solution $x \in \mathbb{F}^n$ for the simultaneous system of equations in the polynomial vector $\mathcal{P}$ and given $y \in \mathbb{F}^m$ is called an SME-problem, cf Figure 2.1.

$$\boxed{\left\{\begin{array}{rcl} y_1 & = & p_1(x_1, \ldots, x_n) \\ y_2 & = & p_2(x_1, \ldots, x_n) \\ & \vdots & \\ y_m & = & p_m(x_1, \ldots, x_n) \end{array}\right.}$$

Figure 2.1: Example of an SME-problem with n variables and m equations

The key-length in a system based on the intractability of the simultaneous solving of multivariate, non-linear equations (*i.e.*, $d \geq 2$) can be computed using the following formulas. Therefore, we first define

$$\tau_{(d)}(\mathbb{F}^n) := \left\{ \begin{array}{ll} \sum_i^{\min(|\mathbb{F}|-1,d)} \binom{n}{i} & \text{for } d > 0 \\ 1 & \text{for } d = 0 \end{array} \right.$$

for the number of terms in n variables of degree d over the finite field $\mathbb{F}$. For the correctness of the above formula, we notice that we have $x^{q-1} = 1$ with $q := |\mathbb{F}|$ in all finite fields (cf Lemma 2.1.5). Using this, we can write

$$\tau^d(\mathbb{F}^n) := \sum_{i=0}^{d} \tau_{(i)}(\mathbb{F}^n)$$

for the number of all terms in a single polynomial equation over $\mathbb{F}$ with maximal degree d and in n variables.

In particular, this leads to the following size function for given parameters $\mathbb{F}, n, m, d, q := |\mathbb{F}|$:

$$\mathit{size}(\mathbb{F}, n, m, d) := m\tau^d(\mathbb{F}^n) \log_2 q \ . \tag{2.1}$$

In general, we obtain a key-length of $O(mn^d)$ for the public key — or $O(n^{d+1})$ for $m = n$.

2.2.2 $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Polynomials

For any q and $d = 2$, we speak about the problem of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations and denote the class of corresponding polynomial vectors $\mathcal{P}$ with $\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ (cf Figure 2.1 for the general case). As we will see below, this class plays an important role for the construction of public key schemes based on the problem of polynomial equations over finite fields. Therefore, we state the polynomials p_i explicitly for this case:

$$\begin{array}{rcl} p_1(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{1,j,k}x_jx_k + \sum_{j=1}^{n} \beta_{1,j}x_j + \alpha_1 \\ & \vdots & \\ p_i(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{i,j,k}x_jx_k + \sum_{j=1}^{n} \beta_{i,j}x_j + \alpha_i \\ & \vdots & \\ p_m(x_1,\ldots,x_n) & := & \displaystyle\sum_{1\le j\le k\le n} \gamma_{m,j,k}x_jx_k + \sum_{j=1}^{n} \beta_{m,j}x_j + \alpha_m \end{array}$$

with the coefficients $\gamma_{i,j,k}, \beta_{i,j}, \alpha_i \in \mathbb{F}$. In the case of $d = 2$, we call them quadratic ($\gamma_{i,j,k}$), linear ($\beta_{i,j}$), and constant (α_i) coefficients, respectively. In short, we write this polynomial vector as $\mathcal{P} := (p_1, \ldots, p_m)$ and moreover have $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$. By convention, we require $j < k$ for $q = 2$ as $x_i^2 = x_i$ in GF(2). If the number of variables is equal to the number of equations, we write $\mathcal{MQ}(\mathbb{F}^n)$ for brevity rather than $\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^n)$.

In addition, we state the formula for the number of terms for *one* polynomial of the $\mathcal{MQ}$-problem explicitly:

$$\tau(n) \;:=\; \begin{cases} 1 + n + \frac{n(n-1)}{2} = 1 + \frac{n(n+1)}{2} & \text{if } \mathbb{F} = GF(2) \\ 1 + n + \frac{n(n+1)}{2} = 1 + \frac{n(n+3)}{2} & \text{otherwise .} \end{cases} \tag{2.2}$$

The above formula assumes polynomials with quadratic and linear terms plus a constant term.

The prominent role of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations is easily seen by the two following observations: first, the public key size increases with $O(mn^d)$ — and is hence very sensitive to the degree d. Therefore, we want d to be as small as possible. On the other hand, solving quadratic systems is already $\mathcal{NP}$-complete and also hard on average. We refer the reader to cf [GJ79, p. 251] and [PG97, App.]. A detailed proof can be found in Section 2.5.

2.2.3 Affine Transformations

As we will see in the following sections, affine transformations play an important role in the theory of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. Hence, to have all necessary tools at hand in the sequel, we review some of their properties. In this context, the following lemma proves useful.

Remark 2.2.1 *In the remainder of this text, we denote elements of vector spaces by small letters and elements of extension fields by capital letters,* e.g., *vector $x \in \mathbb{F}^n$ but element $X \in \mathbb{E}$.*

Lemma 2.2.2 *Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements. Then we have $\prod_{i=0}^{n-1}\left(q^n - q^i\right)$ invertible $(n \times n)$-matrices over $\mathbb{F}$.*

Proof. We observe that we have full choice for the first row vector of our matrix — except the zero-vector. With an inductive argument we see that we have full choice for each consecutive row vector — except the span of the previous row vectors. Hence, we have a total of $\left(q^n - q^{j-1}\right)$ independent choices for the j^{th} row vector. □

Next, we recall some basic properties of affine transformations over the finite field $\mathbb{F}$ and its n^{th}-degree extension $\mathbb{E}$.

Definition 2.2.3 *Let $M_S \in \mathbb{F}^{n\times n}$ be an $(n \times n)$ matrix and $v_s \in \mathbb{F}^n$ a vector and let $S(x) := M_S x + v_s$. We call this the "matrix representation" of the affine transformation S.*

Definition 2.2.4 *Moreover, let $s_1, \ldots, s_n$ be n polynomials of degree 1 at most over $\mathbb{F}$,* i.e., *we have $s_i(x_1, \ldots, x_n) := \beta_{i,1}x_1 + \ldots + \beta_{i,n}x_n + \alpha_i$ with $1 \leq i, j \leq n$ and $\alpha_i, \beta_{i,j} \in \mathbb{F}$. Let $S(x) := (s_1(x), \ldots, s_n(x))$ for $x := (x_1, \ldots, x_n)$ a vector over $\mathbb{F}^n$. We call this the "multivariate representation" of the affine transformation S.*

Remark 2.2.5 *The multivariate and the matrix representation of an affine transformation S are interchangeable. We only need to identify the corresponding coefficients: $(M_S)_{i,j} \leftrightarrow \beta_{i,j}$ and $(v_S)_i \leftrightarrow \alpha_i$ for $1 \leq i, j \leq n$.*

In addition, we can also use the "univariate representation" over the extension field $\mathbb{E}$ of the transformation S.

Definition 2.2.6 *Let $0 \leq i < n$ and $A, B_i \in \mathbb{E}$. Then we call the polynomial $S(X) := \sum_{i=0}^{n-1} B_i X^{q^i} + A$ the "univariate representation" of the affine transformation $S(X)$.*

The important point here is that $x \to x^q$ is a linear mapping in the finite field $\mathbb{F}$ and also its extension field $\mathbb{E}$, cf Lemma 2.1.5. Hence, all sums which have only powers of the form x^{q^i} for $0 \le i < n$ in $\mathbb{E}$ are also linear mappings. A proof of this statement can be found, *e.g.*, in [KS99].

Lemma 2.2.7 *An affine transformation in univariate representation can be efficiently transfered into matrix representation and vice versa.*

Proof. As we already know that both the univariate and the matrix representation exist, it is sufficient to give an algorithm to transfer an affine transformation given in one of these representations to the other representation.

We start with the univariate polynomial $P(X) := \sum_{i=0}^{n-1} B_i X^{q^i} + A$ for given $B_i, A \in \mathbb{E}$ and compute a corresponding matrix $M \in \mathbb{F}^{n \times n}$ and a vector $v \in \mathbb{F}^n$. For this purpose, we define $\eta_0 \in \mathbb{F}^n$ the 0 vector, and $\eta_i \in \mathbb{F}^n : 1 \le i \le n$, a vector with its i^{th} coefficient 1, the others 0. Moreover, we use the canonical bijection $\phi : \mathbb{E} \to \mathbb{F}^n$, cf Definition 2.1.6. Now, we compute $v := \phi(P(\phi^{-1}(\eta_0)))$, and $M_i := \phi(P(\phi^{-1}(\eta_i)))$ where the vector $M_i \in \mathbb{F}^n$ denotes the i^{th} row of the matrix M. By construction, we have $\phi(P(X)) = M.\phi(X) + v$ for all $X \in \mathbb{E}$.

The converse computation, *i.e.*, to obtain a polynomial $P \in \mathbb{E}[x]$ of the required form for given matrix $M \in \mathbb{F}^{n \times n}$ and a vector $v \in \mathbb{F}^n$ is a little more difficult. Note that the polynomial P is very sparse as it has only $(n+1)$ non-zero coefficients. We start with the observation $\phi^{-1}(M.0 + v) = P(0) = A$, *i.e.*, we have $A := \phi^{-1}(v)$. For the coefficients $B_0, \ldots, B_{n-1} \in \mathbb{E}$, it is sufficient to solve the following matrix equation for *given* $X_i \in \mathbb{E}, 1 \le i \le \lambda$ and $\lambda \ge n$ over the extension field $\mathbb{E}$:

$$\begin{pmatrix} X_1^{q^0} & X_1^{q^1} & \ldots & X_1^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ X_i^{q^0} & X_i^{q^1} & \ldots & X_i^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ X_\lambda^{q^0} & X_\lambda^{q^1} & \ldots & X_\lambda^{q^{n-1}} \end{pmatrix} \begin{pmatrix} B_0 \\ B_1 \\ \vdots \\ \vdots \\ B_{n-1} \end{pmatrix} = \begin{pmatrix} M.\phi(X_1) \\ M.\phi(X_2) \\ M.\phi(X_3) \\ \vdots \\ \vdots \\ M.\phi(X_\lambda) \end{pmatrix} \qquad (2.3)$$

To obtain a unique solution, we need the rank of the above matrix to be equal to n. As it is a $(\lambda \times n)$-matrix, we can be sure that the rank will not exceed n. Moreover, if the rank is smaller than n, we increase the value of λ, until the matrix has full rank and we obtain a unique solution. This is possible as we have full control on X and hence can make sure that all rows are linearly independent. We do not expect $\lambda \gg n$ in practice; this was confirmed through simulations.

We note that all computations in this proof can be done in polynomial time: we need matrix multiplications in the ground field $\mathbb{F}$, and Gauss operations in the extension field $\mathbb{E}$ to solve the above linear equation. Hence, we can transfer efficiently between both representations. $\square$

Remark 2.2.8 *The matrix from (2.3) resembles Vandermonde matrices. However, there is an important difference: rows of Vandermonde matrices have the form α^i for $i = 0, \ldots, (n-1)$ while the matrix from (2.3) uses double exponents. While only a small change at first glance, it changes the structure of the matrix. For example, a Vandermonde matrix has only 1's in its first column while the above matrix may have any value from $\mathbb{E}$ here. As a consequence, the machinery developed for Vandermonde matrices cannot be applied in the proof of Lemma 2.2.7.*

In the remainder of this thesis, we denote the class of *affine* transformations $\mathbb{F}^n \to \mathbb{F}^n$ by $\mathrm{Aff}_0(\mathbb{F}^n)$, and the class of linear transformations, *i.e.*, with the constant term equal to 0, will be denoted by $\mathrm{Hom}_0(\mathbb{F}^n)$ for *homomorphism.* In both cases, the subscript $_0$ indicates that the all zero transformation is also included in this set. Moreover, for affine and linear transformations, we can use the matrix representation to determine if the corresponding transformation is a bijection or not. For a bijection, the matrix M_S needs to have full rank. In most cases, we will use bijections in this thesis and hence, use $\mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $\mathrm{Hom}^{-1}(\mathbb{F}^n)$, *i.e.*, the class of *invertible* affine and linear transformations, respectively. In this context we want to stress that $(\mathrm{Aff}^{-1}(\mathbb{F}^n), \circ)$ and $(\mathrm{Hom}^{-1}(\mathbb{F}^n), \circ)$ form groups for the symbol "$\circ$" being function composition.

At some points, we will not only need transformations within the same vector space, but transformations $S(x) : \mathbb{F}^n \to \mathbb{F}^m$ with $n \neq m$. We therefore extend our notation to $\mathrm{Aff}_0(\mathbb{F}^n, \mathbb{F}^m)$ and $\mathrm{Hom}_0(\mathbb{F}^n, \mathbb{F}^m)$ in this case. Obviously, the corresponding transformations cannot be bijective anymore, but injective in the case $n < m$ and surjective in the case $n > m$. As in the case of bijective transformations, we can use the rank of the corresponding matrix to determine if a given transformation is injective or surjective. By abusing the notation from above, we write $\mathrm{Aff}^{-1}(\mathbb{F}^n, \mathbb{F}^m)$ and $\mathrm{Hom}^{-1}(\mathbb{F}^n, \mathbb{F}^m)$ in this case, *i.e.*, both for the surjective $(n > m)$ and the injective $(n < m)$ case. In any case, function composition is no longer defined on these objects and hence, $(\mathrm{Aff}^{-1}(\mathbb{F}^n, \mathbb{F}^m), \circ)$ $(\mathrm{Hom}^{-1}(\mathbb{F}^n, \mathbb{F}^m), \circ)$ are no groups anymore.

2.2.4 Matrix Representation

As we saw above, it is possible to represent affine transformations in three different ways: univariate, multivariate, and as a matrix. Here, we see that one can also express $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials as square matrices over the ground field $\mathbb{F}$. Let

$$p_i = \sum_{1 \leq j \leq k \leq n} \gamma_{i,j,k} x_j x_k + \sum_{j=1}^{n} \beta_{i,j} x_j + \alpha_i$$

be a public key polynomial as defined in Section 2.2.2, *i.e.*, with the public key coefficients $\alpha_i, \beta_{i,j}, \gamma_{i,j,k} \in \mathbb{F}$ and the unknowns $x_i \in \mathbb{F}$. In order to uniquely express its homogeneous quadratic parts, *i.e.*, the coefficients $\gamma_{i,j,k}$ for $j \neq k$ in a symmetric matrix $P_i \in \mathbb{F}^{n \times n}$, we need to distinguish odd and even characteristic.

$$\boxed{P_i = \begin{pmatrix} \gamma_{i,1,1} & \gamma_{i,1,2}/2 & \cdots & \cdots & \gamma_{i,1,n}/2 \\ \gamma_{i,1,2}/2 & \gamma_{i,2,2} & & & \gamma_{i,2,n}/2 \\ \vdots & \vdots & \ddots & & \vdots \\ \gamma_{i,1,n-1}/2 & \gamma_{i,2,n-1}/2 & & \gamma_{i,n-1,n-1} & \gamma_{i,n-1,n}/2 \\ \gamma_{i,1,n}/2 & \gamma_{i,2,n}/2 & \cdots & \gamma_{i,n-1,n}/2 & \gamma_{i,n,n} \end{pmatrix}}$$

Figure 2.2: Matrix Representation P_i of the Public Key p_i

- For odd characteristics, the matrix elements $(P_i)_{a,b}$ on row a and column b of the symmetric matrix P_i are determined by

$$\begin{cases} (P_i)_{a,b} := (P_i)_{b,a} := \frac{\gamma_{i,a,b}}{2} & \text{for } 1 \leq a < b \leq n \\ (P_i)_{a,a} := \gamma_{i,a,a} & \text{for } 1 \leq a \leq n\,. \end{cases}$$

  So, instead of evaluating the quadratic parts of p_i by the vector x, we may also perform xP_ix^t as matrix-vector multiplications (here ^{t} denotes transposition), cf Figure 2.2 for a graphical representation of this idea.

- For even characteristic, division by 2 is not defined. Therefore the form $P_i := L_i + L_i^t$ for lower triangular matrices L_i is used. This way, we loose the quadratic coefficients $\gamma_{i,i}$ of the public polynomials. However, in characteristic 2, these quadratic terms are linear and we can therefore ignore them. To the knowledge of the author, the above observation has been initially reported in [KPG99] and is there credited to *Don Coppersmith*.

Although it seems strange at first glance that such a matrix representation of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials could be useful, we will see in Chapter 4 that it is crucial for certain types of attacks.

2.3 $\mathcal{MQ}$-trapdoor

To be useful for public key cryptology, we do not only need an intractable problem, but also a way of embedding a trapdoor into it. For the $\mathcal{MQ}$-problem as stated in Section 2.2, we are able to embed a trapdoor $(S, \mathcal{P}', T)$ into a system of

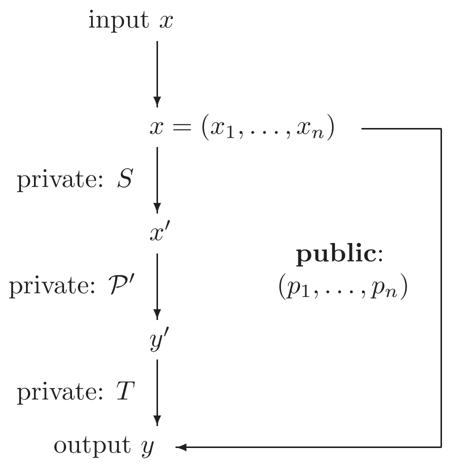

Figure 2.3: Graphical Representation of the $\mathcal{MQ}$-trapdoor $(S, \mathcal{P}', T)$

equations $\mathcal{P}$, cf Figure 2.3. Here we have $(S, \mathcal{P}', T) \in \mathrm{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathrm{Aff}^{-1}(\mathbb{F}^m)$ and $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$, *i.e.*, S is an invertible affine transformation $S : \mathbb{F}^n \to \mathbb{F}^n$ and T is an invertible affine transformation $T : \mathbb{F}^m \to \mathbb{F}^m$. Moreover, $\mathcal{P}'$ is a polynomial vector as defined in Section 2.2, *i.e.*, all m polynomials in n variables each have degree $d = 2$. In particular, we have $\mathcal{P}'$ as a function $\mathcal{P}' : \mathbb{F}^n \to \mathbb{F}^m$. In the remainder of this thesis, we denote components of the hidden quadratic transformation $\mathcal{P}'$ by a prime $'$, *e.g.*, the variables $x'_1, \ldots, x'_n$ or the coefficients $\alpha'_i, \beta'_{i,j}, \gamma'_{i,j,k}$ for $1 \leq i \leq m$ and $1 \leq j \leq k \leq n$. In general, $\mathcal{P}'$ consists of non-homogeneous polynomials, *i.e.*, we have at least one non-zero $\beta'_{i,j}$ and one non-zero α'_i in this polynomial-vector.

We want to point out that the trapdoor from Figure 2.3 is the only one possible: as we restricted our attention to $\mathcal{M}$ultivariate $\mathcal{Q}$*uadratic* equations, we cannot have a degree higher than 2 for the public key equations. But this implies immediately that we can have at most one degree 2 transformation in the overall construction. Hence, all variations of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems (cf Section 3.2) can only use degree one equations for the two affine transformations S, T and some hidden invertible *quadratic* system of polynomials for $\mathcal{P}'$. In particular, these two affine transformations are used to hide the internal structure of the central equations $\mathcal{P}'$ from the eyes of an attacker. This is necessary as we need the central map $\mathcal{P}'$ to be invertible in contrast to the public key $\mathcal{P}$ alone.

Another way of "modifying" the above trapdoor is the use of several affine transformations. However, as we noted in the previous section, this does not help as affine transformations form a group and hence, are closed under composition.

2.3.1 Signature Verification

Signature verification is the same for all schemes based on the difficulty of the $\mathcal{MQ}$-problem: evaluate the polynomial vector $\mathcal{P}$ for a given signature $x \in \mathbb{F}^n$. If the result is the same as the given message vector $y \in \mathbb{F}^m$, we accept the signature, otherwise we reject. For short, we write $y \stackrel{?}{=} \mathcal{P}(x)$ where $\stackrel{?}{=}$ denotes comparison. So we perform the following m checks of elements in $\mathbb{F}$:

$$\begin{aligned} y_1 &\stackrel{?}{=} p_1(x_1, \ldots, x_n) \\ &\vdots \\ y_m &\stackrel{?}{=} p_m(x_1, \ldots, x_n) \end{aligned}$$

As each polynomial has $\tau(n) = O(n^2)$ coefficients (cf Section 2.2), such an evaluation takes a total of $O(mn^2)$ multiplications and additions in the finite field $\mathbb{F}$. Strategies for fast evaluation of the public key are discussed in [CGP01, CGP03a].

2.3.2 Signature Generation

To generate a signature, we make use of the private key, *i.e.*, the $\mathcal{MQ}$-trapdoor $(S, \mathcal{P}', T) \in \mathrm{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathrm{Aff}^{-1}(\mathbb{F}^m)$. Here we observe that we need to invert each individual step, *i.e.*, we need to compute the vector $y' := T^{-1}(y)$ for given y, followed by $x' := \mathcal{P}'^{-1}(y')$, and finally $x := S^{-1}(x')$, cf Figure 2.4.

We start with the inversion of $S(x)$: as we saw in Section 2.2.3, we can write this affine transformation using an invertible matrix $M \in \mathbb{F}^{n \times n}$ and a vector $v \in \mathbb{F}^n$, *i.e.*, we have $S(x) := Mx + v$. Therefore, its inverse is given by the affine transformation $S^{-1}(x) := M^{-1}(x - v)$. Similar, we can invert the second affine transformation T.

Things are more complicated for the system of polynomials $\mathcal{P}'$ as inversion strategies differ for individual trapdoor functions, *e.g.*, MIA, HFE, STS, or UOV. Therefore, we will discuss the inversion strategy in the individual sections (cf Section 3.1). However, we want to stress that it is enough to find *one* pre-image of $\mathcal{P}$ to obtain a valid signature, *i.e.*, we only need one $x' \in \mathbb{F}^n$ with $\mathcal{P}'(x') = y'$ for given $y' \in \mathbb{F}^m$. In case $\mathcal{P}' : \mathbb{F}^n \to \mathbb{F}^m$ and hence $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ is not a surjection, we add some random bits to the input $x \in \mathbb{F}^n$, cf [CGP01] for an

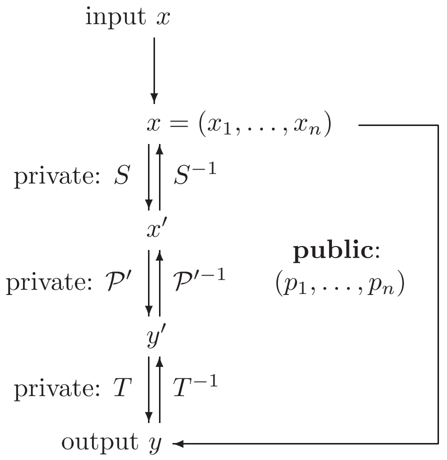

Figure 2.4: Inverting the $\mathcal{MQ}$-trapdoor

outline of this idea. In a nutshell, even a small number of random bits ensures that we obtain a valid signature in all practical cases. In [CGP01], it is shown that we need only 7 random bits to obtain a valid signature with probability $1 - 2^{-187}$ for the HFE trapdoor for a given message $x \in \mathbb{F}^n$. As we will see in the next section, matters are slightly more complicated for decryption.

2.3.3 Decryption

Decryption and signature generation are quite similar — except for the fact that we usually need to compute *all* possible pre-images $X'_1, \ldots, X'_k \in \mathbb{F}^n$ which satisfy the equation $\mathcal{P}'(X') = y'$ for given $y' \in \mathbb{F}^m$ and some $k \in \mathbb{N}$. Depending on the scheme used, it may happen that we do not have a unique solution X' for this equation. Hence, we need a possibility to pick the right X_i from the set of all possible solutions $Q := \{X_i \in \mathbb{F}^n : X_i := S^{-1}(X'_i) \text{ for } 1 \leq i \leq k\}$. Assuming that we decrypt a valid ciphertext, we have Q non-empty and hence, $k \in \mathbb{Z} : k \geq 1$.

This problem has been discussed in [Pat96b] and its author suggests to use either error-correcting codes or a cryptographically secure hash function to solve it. To our knowledge, only hash functions have been used so far in this context. Denote such a hash function $H(\cdot) : \mathbb{F}^n \to \{0,1\}^h$ where h is the length of the hash string. Hence, during encryption (see below), the hash-value $\tilde{x} := H(x)$ is computed and then used to pick the right X_i from the set Q by simply checking if the corresponding hashes match, *i.e.*, if we have $H(X_i) \stackrel{?}{=} \tilde{x}$. As the hash function

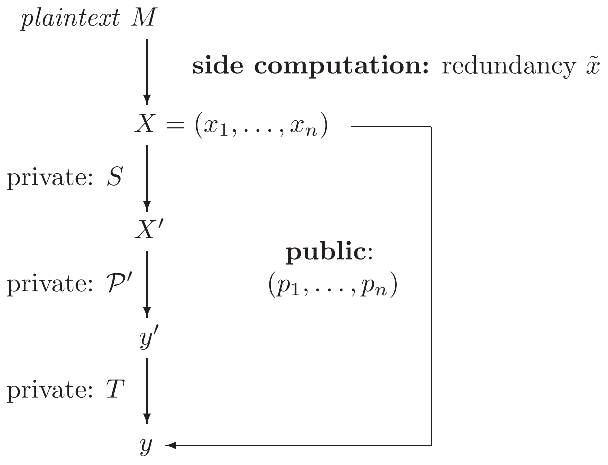

Figure 2.5: $\mathcal{MQ}$-systems for Encryption of Message M with Ciphertext $(y, \tilde{x})$

is assumed to be cryptographically secure, an attacker cannot use the knowledge of $\tilde{x}$ to gain an advantage when computing x. In this context we want to point out that the attacker can always verify his guesses on x by checking if $y = \mathcal{P}(x')$ match for some guess $x' \in \mathbb{F}^n$. However, the workload of this procedure clearly depends on the size of the set Q: on average, we will need to check $|Q|/2$ elements before finding the right X_i. Hence, Q may not be too large in practice. As we will see below, this is a serious obstacle for constructing a secure and efficient encryption scheme based on the $\mathcal{MQ}$-problem.

In this context, the question of the optimal length of the output of such a hash function is also important: having the corresponding hash too short, we may not be able to find a unique x_i from the set Q. Having h too long, we waste bandwidth. This question has been elaborated in [Dau01, Section 2.3.3]. In a nutshell, we need an 80 bit hash result to have a probability of $1 - 2^{-80}$ for unique decryption. More general, we need h bits to have a probability of $1 - 2^{-h}$ for unique deciphering.

Using the idea of an error correcting code on x, *i.e.*, to encode a message M using some code word x, and only accepting elements from Q which are a correct codeword, does not seem to have advantages over the idea of using hash-functions as it enforces a bigger parameter n: our new message space is now the message space of the error correcting code while the size of the code space determines the number of input variables n. Hence, the number of coefficients increases by the message expansion of the error correcting code used and therefore also the time for decryption or encryption. Moreover, having redundancy in the clear text is usually not a good idea as it may be exploited in an attack. We therefore do not advise this strategy but encourage the use of hash-functions in this context.

A similar strategy is padding: here, the first $f < n$ bits of the message vector $X \in \mathbb{F}^n$ are fixed to some values $v_1, \ldots, v_f \in \mathbb{F}$ and only signatures with $x_1 = v_1, \ldots, x_f = v_f$ are accepted. Hence, we have the same concerns as for methods using error correcting codes: we need a bigger parameter n for secure schemes but may give the adversary an advantage as he already knows parts of the message. Assuming that we want to filter out wrong signatures with probability $1 - 2^{-s}$, we need $f := \log_2 q$ padding bits. A typical value for s would be 80.

Remark 2.3.1 *To use $\mathcal{MQ}$-systems in real-world applications, the question of semantic security becomes pressing. This means that an attacker cannot use the encrypted text $y' \in \mathbb{F}^m$ to gain information about the original message $x \in \mathbb{F}^n$. Hence, the question of suitable padding schemes arises naturally.*

2.3.4 Encryption

As discussed in the previous section, the function $\mathcal{P}'(x') = y'$ is usually not surjective — and consequently, neither is $\mathcal{P}(x) = y$. Hence, we need to compute some redundancy to allow unique decryption, cf Figure 2.5. Consequently, encryption consists of two steps: first, we evaluate the public key and second, we compute this redundancy $\tilde{x}$:

1. $y := \mathcal{P}(x)$
2. $\tilde{x} := H(x)$

for some hash function $H(\cdot)$, cf previous section. The encrypted message now consists of the pair $(y, \tilde{x}) \in \mathbb{F}^m \times \{0,1\}^h$ for $h \in \mathbb{N}$ being the length of the hash-string used. In contrast to decryption, encryption is always unique as there exists only one $y \in \mathbb{F}^m$ for any given $x \in \mathbb{F}^n$.

2.4 Univariate and Multivariate Representations

After outlining the general structure of the $\mathcal{MQ}$-trapdoor, we move on to the multivariate representation of univariate functions.

Therefore, we need to come back to Definition 2.1.6 which allowed us to transfer elements between the extension field $\mathbb{E}$ and the vector space $\mathbb{F}^n$, using the canonical bijection $\phi : \mathbb{E} \to \mathbb{F}^n$ and its inverse $\phi^{-1} : \mathbb{F}^n \to \mathbb{E}$. With this definition, we are now able to formally prove an important lemma in the context of $\mathcal{MQ}$-systems.

Lemma 2.4.1 *Let $\mathbb{E}$ be an extension field and $\mathbb{F}$ the corresponding ground field. We recall that we have $q := |\mathbb{F}|$ as the number of its elements. In addition, let n be the dimension of $\mathbb{E}$ over the ground field. Consider the univariate monomial $P(X) := CX^{q^a+q^b}$ over $\mathbb{E}$ for some $a, b \in \mathbb{N}$ and $C \in \mathbb{E}$. Then there exists a polynomial vector $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n)$ which computes the same function,* i.e., *$\forall\ W \in \mathbb{E} : \phi(P(W)) = \mathcal{P}(\phi(W))$.*

PROOF. First, we decompose $P(X)$ into the two univariate monomials $U(X) := CX^{q^a}$ and $V(X) := X^{q^b}$. Second, we observe that computing in $\mathbb{Z}/(q^n-1)\mathbb{Z}$ allows us to reduce the degree of the monomials $U(X), V(X)$ below q^n. In particular, we can obtain obtain two integers $a', b' \in \mathbb{Z}$ with $0 \leq a', b' < n$ such that $U(X) = U'(X)$ for $U'(X) := CX^{q^{a'}}$ holds for all inputs $X \in \mathbb{E}$. The same is true for $V(X) = V'(X)$ with $V'(X) := X^{q^{b'}}$. Therefore, and w.l.o.g., we can assume $0 \leq a, b < n$.

Next we note that the monomials of U, V are affine transformations in univariate representation, *i.e.*, we can apply Lemma 2.2.7 to obtain the corresponding multivariate representations $\mathcal{U}$ and $\mathcal{V}$. Denoting the components of the polynomial vector $\mathcal{U}$ by $u_1, \ldots, u_n$ we can now write

$$\begin{aligned}\mathcal{U}(x_1,\ldots,x_n) &= \phi(U(\phi^{-1}(x_1,\ldots,x_n)))\\ &= \phi(u_1(x_1,\ldots,x_n)\\ &\quad + tu_2(x_1,\ldots,x_n)\\ &\quad + \ldots\\ &\quad + t^{n-1}u_n(x_1,\ldots,x_n))\,.\end{aligned}$$

Similar, we obtain a mixed $\mathbb{F}^n/\mathbb{E}$-representation of the polynomial vector $\mathcal{V}$. Multiplying $\mathcal{U}, \mathcal{V}$ in $\mathbb{E}$, *i.e.*, in particular, modulo the irreducible defining polynomial $i(t)$, yields the corresponding $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials by construction. □

Remark 2.4.2 *Instead of computing the multivariate polynomials as outlined in the above proof, we can also use multivariate polynomial interpolation, cf [MI88, Wol04] for details.*

Corollary 2.4.3 *For a polynomial of the form*

$$P(X) := \sum_{\substack{0\leq i,j\leq D\\ q^i+q^j\leq D}} C_{i,j}X^{q^i+q^j} \text{ with } C_{i,j} \in \mathbb{E}$$

with $D \in \mathbb{N}$ and $D < q^n$, there exists a polynomial vector $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n)$ which computes the same function.

Lemma 2.4.4 *Let $\mathbb{F}$ be a ground field and $\mathbb{E}$ an n-dimensional extension of $\mathbb{F}$. Then for the polynomial*

$$P(X) := \sum_{0\le i\le j<n} C_{i,j} X^{q^i+q^j} + \sum_{i=0}^{n-1} B_i X^{q^i} + A$$

with coefficients $C_{i,j}, B_i, A \in \mathbb{E}$ there exists a unique multivariate quadratic polynomial vector $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n)$, not necessarily of full degree, which computes the same function, i.e., *we have $P(X) = \phi^{-1}(\mathcal{P}(\phi(X)))\ \forall X \in \mathbb{E}$.*

PROOF. We use Corollary 2.4.3 for the quadratic terms, Lemma 2.2.7 on the affine part, and add up the result. □

Interestingly, the converse is also true:

Lemma 2.4.5 *Let $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n)$ be a $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations and $\mathbb{E}$ an n-dimensional extension of the ground field $\mathbb{F}$. Then there exists a unique univariate polynomial*

$$P(X) := \sum_{0\le i\le j<n} C_{i,j} X^{q^i+q^j} + \sum_{i=0}^{n-1} B_i X^{q^i} + A$$

with coefficients $C_{i,j}, B_i, A \in \mathbb{E}$ which computes the same function as $\mathcal{P}$, i.e., *we have $P(X) = \phi^{-1}(\mathcal{P}(\phi(X)))\ \forall X \in \mathbb{E}$.*

PROOF. We use a counting argument and will assume $\mathbb{F} \neq$GF(2) for simplicity. Consider all polynomials $P(X) \in \mathbb{E}[X]$ which have the above form. They have $\binom{n}{2}+n+n+1$ coefficients in total: the quadratics (coefficients $C_{i,j}$ with $i \neq j$), the quadratics in the same variable (coefficients $C_{i,i}$), the linear terms (coefficients B_i) and the constant term A. Hence, there are $q^{n\cdot\left(\frac{n(n+3)}{2}+1\right)}$ choices in total for these polynomials $P(X)$. On the other hand, we know from (2.2) that we have a total choice of $q^{n\cdot\tau(n)} = q^{n\cdot\left(\frac{n(n+3)}{2}+1\right)}$ for polynomial vectors in $\mathcal{MQ}(\mathbb{F}^n)$. In addition, Lemma 2.4.4 shows that each of these polynomials $P(X)$ has a unique multivariate representation, denoted $\mathcal{P}(X)$. Moreover, both functions compute the same output for any given input $X \in \mathbb{E}$. This is not true for two polynomials $P_1(X), P_2(X) \in \mathbb{E}[X]$, $P_1 \neq P_2$ and their corresponding polynomial vectors $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{MQ}(\mathbb{F}^n)$. Hence, using the counting from above we are able to conclude that for each polynomial vector $\mathcal{P}$, there is one unique univariate polynomial $P(X)$. This completes the proof for the case $\mathbb{F} \neq$ GF(2).

The same proof runs through for GF(2), but we have to adjust our counting slightly as $x_i^2 = x_i$ holds for $1 \le i \le n$ and consequently no terms of the form $X^{q^i+q^i}$ in the polynomial $P(X)$. However, the overall idea remains the same. □

Remark 2.4.6 *The previous lemma has already been shown in a more general setting in [KS99, Lemma 3.3]; in this thesis, the proof has been simplified for the case of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations. Another proof of this lemma, but this time restricted to the case $\mathbb{F} = GF(2)$, can be found in [MIHM85]. Moreover, the univariate representation of multivariate quadratic equations can be computed efficiently: we use polynomial interpolation on a total of $O(n^2)$ points from $\mathbb{E}$, which translates to $O(n^3)$ elements from $\mathbb{F}$, cf Lemma 2.2.7 for the general idea.*

Lemma 2.4.7 *Let $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ be a $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations with $n \in \mathbb{N}$ variables and $m \in \mathbb{N}$ equations. For the cases (a) $m < n$ and (b) $m > n$ there exists a univariate representation $P \in \mathbb{E}[X]$ for $\mathbb{E}$ being an (a) n-dimensional and (b) m-dimensional extension of the ground field $\mathbb{F}$.*

Proof. Case *(a)*: We have $m < n$ and $\mathbb{E}$ an n-dimensional extension of the ground field $\mathbb{F}$ and the canonical bijection $\phi : \mathbb{E} \to \mathbb{F}^n$ (see Definition 2.1.6). Moreover, consider the reduction/projection transformation $R : \mathbb{F}^n \to \mathbb{F}^m$ defined as

$$R(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) := (x_1, \ldots, x_m)$$

and its "inverse" transformation $R^{-1} : \mathbb{F}^m \to \mathbb{F}^n$ which is defined as

$$R^{-1}(x_1, \ldots, x_m) := (x_1, \ldots, x_m, 0, \ldots, 0)\,.$$

Using Lemma 2.4.5, we compute a polynomial $P \in \mathbb{E}[x]$ with

$$P(X) = \phi^{-1}(R^{-1}(\mathcal{P}(\phi(X))))\ \forall X \in \mathbb{E}\,.$$

By construction, we have

$$R(\phi(P(\phi^{-1}(x)))) = \mathcal{P}(x)\ \forall x \in \mathbb{F}^n\,.$$

An alternative way of writing the above statement is to replace the "inverse reduction" $R^{-1}(X)$ by adding zero polynomials $p_{m+1}, \ldots, p_n$ to the multivariate function $\mathcal{P}$, *i.e.*, these polynomials are all chosen to be the zero polynomial. This way, we obtain $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^n$ and can therefore apply Lemma 2.4.5 directly.
Case *(b)*: We have $m > n$ and $\mathbb{E}$ an m-dimensional extension of the ground field $\mathbb{F}$ and define $\phi : \mathbb{E} \to \mathbb{F}^m$. Moreover, consider the reduction transformation $R : \mathbb{F}^m \to \mathbb{F}^n$ defined as $R(x_1, \ldots, x_n, x_{n+1}, \ldots, x_m) := (x_1, \ldots, x_n)$. Using Lemma 2.4.5, we compute a polynomial $P \in \mathbb{E}[x]$ with

$$P(X) = \phi^{-1}(\mathcal{P}(R(\phi(X))))\ \forall X \in \mathbb{E}\,.$$

By construction, this polynomial computes the required function. Moreover, due to the definition of the reduction function $R(x)$, the degree of $\mathcal{P}(R(x))$ remains quadratic. □

Remark 2.4.8 *Due to their construction, both polynomials $P(X)$ in (a) and (b) of Lemma 2.4.7 are a univariate representation of the corresponding $\mathcal{P}(x)$. For a fixed reduction transformation R and a fixed extension field $\mathbb{E}$, this univariate polynomial $P(X)$ is even unique.*

Theorem 2.4.9 *Let $n, m \in \mathbb{N}$ and $\mathbb{F}$ a finite field with $q := |\mathbb{F}|$ elements. Moreover, define $k := \max\{n, m\}$ and an extension field $\mathbb{E} := GF(q^k)$. Then there exists a unique univariate representation $P \in \mathbb{E}[X]$ for each multivariate system of equations $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ and vice versa.*

Proof. We use lemmata 2.4.4, 2.4.5, and 2.4.7. □

Remark 2.4.10 *For d the maximal degree of the multivariate equations, and a univariate polynomial $P(X)$ with monomials of the form $X^{q^{i_1}+\cdots+q^{i_{d'}}}$ with $d' \leq d$, it is possible to prove a generalisation of Lemma 2.2.7 by induction over d. Similar, we can show the converse for general polynomials P. However, as this thesis concentrates on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations, we omit the corresponding proofs and refer the reader to [KS99].*

2.5 $\mathcal{NP}$-Completeness of $\mathcal{MQ}$

As we saw in the previous section, $\mathcal{MQ}$ is a quite general problem which can be used for signing and encrypting with an embedded trapdoor. Still, for a one-way trapdoor function, we need two properties: trapdoor *and* one-wayness. To motivate the second, we establish in this section that $\mathcal{MQ}$ is an $\mathcal{NP}$-complete problem. The fact that $\mathcal{MQ}$ over GF(2) is $\mathcal{NP}$-complete has already been pointed out in [GJ79, p. 251]. For the proof, [GJ79, p. 251] refers to a manuscript from A.S.Fraenkel and Y.Yesha that was unpublished in 1977 and also to "private communication" with L.G.Valliant. The text [FY79], from A.S.Fraenkel and Y.Yesha, was published in 1979, has the same title as the manuscript quoted in [GJ79] and gives a proof for the $\mathcal{NP}$-completeness for polynomials over GF(2) and also the algebraic closure of GF(2). It mentions that strictly quadratic polynomials have been treated by L.G.Valiant, without giving any further hint how to prove this. It is called "MinRank" there and also in parts of the older literature as it was seen to be the minimal rank of a matrix. This is similar to the current definition of the MinRank-problem, cf Section 2.6.2 for details. In the newer literature the same problem is usually denoted $\mathcal{MQ}$ ($\mathcal{M}$ultivariate $\mathcal{Q}$uadratic). This section is based on [PG97] which gives a proof both for the case of GF(2) and also for domains, cf Definition 2.5.5 on Page 30. A preliminary version of this section has been published in [Wol02a].

2.5.1 $\mathcal{MQ}$ over GF(2)

To show that $\mathcal{MQ}$ over GF(2) is $\mathcal{NP}$-complete, we first show that it is in $\mathcal{NP}$ and then reduce 3-SAT to $\mathcal{MQ}$.

Definition 2.5.1 *$\mathcal{MQ}$-GF(2): Let $\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ be all systems of quadratic equations over $\mathbb{F} := GF(2)$. Then we call one element $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ an instance of $\mathcal{MQ}$ over GF(2).*

Solvable in $\mathcal{NP}$-time. The following $\mathcal{N}$on-deterministic $\mathcal{P}$olynomial-time algorithm ($\mathcal{NP}$) solves $\mathcal{MQ}$-GF(2) for a given system of equations. The input size can be computed using (2.2) for given finite field $\mathbb{F}$ and natural numbers $n, m \in \mathbb{N}$.

1. Guess an assignment A for $(x_1, \ldots, x_n) \in \{0,1\}^n$.
2. Check if all m equations are satisfied by A.
3. Output `true` or go to an infinity loop, respectively.

As there are m equations, and each equation has $1 + n + \frac{n(n+1)}{2} = \frac{n(n+3)}{2} + 1$ terms at most, Step (2) requires polynomial time. If Step (2) is successful, the algorithm outputs `true`, and terminates but otherwise goes to an infinite loop. So $\mathcal{MQ}$-GF(2) can be solved in $\mathcal{NP}$-time.

Remark 2.5.2 *Note that in the case $\mathbb{F} = GF(2)$, $x^2 = x\ \forall x$. This means that we can have $x_i x_j : i \neq j$ but not $x_i x_i$ as $x_i x_i = x_i^2 = x_i\ \forall i \in \{1, \ldots, n\}$ and x_i over GF(2).*

$\mathcal{NP}$-hardness. In this section, we reduce 3-SAT to $\mathcal{MQ}$-GF(2).

Definition 2.5.3 *3-SAT: Let $B = \{b_1, \ldots, b_n\}$ be a set of Boolean variables, let $L = \{b_1, \overline{b_1}, \ldots, b_n, \overline{b_n}\}$ be the corresponding set of literals, let $c_i \in (L \cup L^2 \cup L^3)$ be clauses of at most 3 literals, and let $C = \{c_1, \ldots, c_m\}$ be a set of these clauses. Then the corresponding 3-SAT problem is to determine if there is an assignment $A \in \{0,1\}^n$ for B such that all c_i are true and hence C is satisfied.*

Using this definition, we show that 3-SAT can be reduced to $\mathcal{MQ}$-GF(2). We start with an instance of 3-SAT. Introduce $X = \{x_1, \ldots, x_n\}$ with x_i being variables over GF(2). Here $\vee$ denotes the Boolean `OR` function and $+$ denotes addition over GF(2). Transfer each clause c_i to equation e_i using the following syntactical transformations where $l_i, l_j, l_k \in L$ and $b_i \in B$:

1. Replace $(l_i \vee l_j \vee l_k)$ by $(l_i + l_j + l_k + l_i l_j + l_i l_k + l_j l_k + l_i l_j l_k)$,

2. Replace $(l_i \vee l_j)$ by $(l_i + l_j + l_i l_j)$.

3. For each variable $b_i \in B$: replace $\overline{b_i}$ with $(1 - x_i)$ and b_i with x_i.

4. Construct an equation $e_i : (c'_i = 1)$ for each transformed clause c'_i.

This algorithm transfers each clause c_i into an equation e_i. Here all equations e_i have at most cubic terms. After expanding and collecting terms, we introduce $\frac{n(n-1)}{2}$ new variables $y_{i,j}$ and $\frac{n(n-1)}{2}$ new equations $y_{i,j} = x_i x_j$, for $i < j$. Moreover, we replace $x_i x_j$ or $x_j x_i$ by $y_{i,j}$ in all equations. This leads to $m + \frac{n(n-1)}{2}$ quadratic equations in $\frac{n(n+1)}{2}$ variables. If there is a solution for this set of equations, we also have a solution for the original 3-SAT problem. As all steps require only polynomial time and space, we reduced 3-SAT in polynomial time to $\mathcal{MQ}$-GF(2), *i.e.*, $\leq_{\text{poly}}$ $\mathcal{MQ}$-GF(2).

Theorem 2.5.4 *$\mathcal{MQ}$-GF(2) is $\mathcal{NP}$-complete.*

Proof. As shown above, MQ-GF(2) $\in \mathcal{NP}$ and 3-SAT $\leq_{\text{poly}}$ MQ-GF(2), which imply that MQ-GF(2) is $\mathcal{NP}$-complete. □

2.5.2 $\mathcal{MQ}$ over Domains

In the previous section, we considered $\mathcal{MQ}$ over GF(2) and proved it to be $\mathcal{NP}$-complete. In this section, we generalise this result for $\mathcal{MQ}$ over domains.

Definition 2.5.5 *$\mathcal{MQ}$-D: Let D be a domain,* i.e., *a commutative ring with 1 and without zero divisors. Then $\mathcal{MQ}$-D is the problem of solving a set of m quadratic equations in n variables over the domain D.*

$\mathcal{NP}$-hardness. We will first reduce $\mathcal{MQ}$-GF(2) to $\mathcal{MQ}$-D, so consider the following set of m equations over GF(2):

$$\sum_{1 \leq j < k \leq n} \gamma_{i,j,k} x_j x_k + \sum_{j=1}^{n} \beta_{i,j} x_j + \alpha_i = 0 \text{ for } (1 \leq i \leq m),$$

where $\gamma_{i,j,k}, \beta_{i,j}, \alpha_i \in$ GF(2) are constants. These m equations form a $\mathcal{MQ}$-GF(2) problem. We transfer each of these m equations to a set of $\frac{n(n+1)}{2}$ equations with $n + \frac{n(n-1)}{2} + (n-1) = \frac{n(n+3)-2}{2}$ variables. This way, we deal with a normal form of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations over the finite field GF(2) which allows to embed operations over GF(2) into the domain D.

$$\begin{cases} \gamma_{i,1,2}x_1x_2 &= y_{i,1,2} \\ \gamma_{i,1,3}x_1x_3 &= y_{i,1,2}+y_{i,1,3} \\ &\vdots \\ \gamma_{i,n-1,n}x_{n-1}x_n &= y_{i,n-2,n}+y_{i,n-1,n} \\ \beta_{i,1}x_1 &= y_{i,n-1,n}+z_{i,1} \\ \beta_{i,2}x_2 &= z_{i,1}+z_{i,2} \\ &\vdots \\ \beta_{i,n-1}x_{n-1} &= z_{i,n-1}+z_{i,n-1} \\ \beta_{i,n}x_n+\alpha_i &= z_{i,n-1} \end{cases}$$

In this system of equations, $\gamma_{i,j,k}, \beta_{i,j}, \alpha_i, x_i$ are the same as above, while $y_{i,j,k}$ and $z_{i,k}$ are new variables over GF(2). So the whole system of m equations and in n variables x_i is transferred to a system of $\frac{mn(n+1)}{2}$ equations with $n+\frac{mn(n-1)}{2}+m(n-1)=\frac{2n+m(n-1)(n+2)}{2}$ variables. This step requires polynomial time in the input, *i.e.*, the initial equations over GF(2).

After this initial step, we transfer the whole system from the finite field GF(2) to the domain D. This can be done by considering each $x_i, y_{i,j,k}, z_{i,j}$ as an element of D and by replacing each $\gamma_{i,j,k}, \beta_{i,j}, \alpha_i \in$ GF(2) by the additive neutral $0 \in D$ or the multiplicative neutral $1 \in D$, respectively. During this transition, multiplication in GF(2) can be replaced by multiplication in D. However, addition of two elements over GF(2) has to be replaced by the following term over the domain D:

$$\begin{array}{rcl} \mathrm{GF}(2) & \to & D \\ (x+y) & \to & (x+y)((1+1)-(x+y)) \end{array}$$

where 1 denotes the multiplicative neutral element in D. So for any $i \in 1, \ldots, n$ we obtain:

$$\begin{cases} \gamma_{i,1,2,k}x_1x_2 &= y_{i,1,2} \\ \gamma_{i,1,3,k}x_1x_3 &= (y_{i,1,2}+y_{i,1,3})((1+1)-(y_{i,1,2}+y_{i,1,3})) \\ &\vdots \\ \gamma_{i,n-1,n}x_{n-1}x_n &= (y_{i,n-2,n}+y_{i,n-1,n})((1+1)-(y_{i,n-2,n}+y_{i,n-1,n})) \\ \beta_{i,1}x_1 &= (y_{i,n-1,n}+z_{i,1})((1+1)-(y_{i,n-1,n}+z_{i,1})) \\ \beta_{i,2}x_2 &= (z_{i,1}+z_{i,2})((1+1)-(z_{i,1}+z_{i,2})) \\ &\vdots \\ \beta_{i,n-1}x_{n-1} &= (z_{i,n-1}+z_{i,n-1})((1+1)-(z_{i,n-1}+z_{i,n-1})) \\ z_{i,n-1} &= (\beta_{i,n}x_n+\alpha_i)((1+1)-(\beta_{i,n}x_n+\alpha_i)) \end{cases}$$

In addition, we have to introduce $\frac{2n+m(n-1)(n+2)}{2}$ new equations (one for each variable) to make sure that we obtain a solution over D if and only if there is a solution over GF(2). Each of these equations has the form

$$x(1-x) = 0.$$

So we transfered a system of m equations and in n variables over GF(2) to a system in $\frac{2n+m(n-1)(n+2)}{2} + \frac{mn(n+1)}{2} = n + m(n^2 + n - 1)$ equations and in $\frac{2n+m(n-1)(n+2)}{2}$ variables over D. All transformations can be done in polynomial time and space and the existence of a solution for the transferred problem implies the existence of a solution for the original one. Hence $\mathcal{MQ}$-GF(2) $\leq_{\text{poly}}$ $\mathcal{MQ}$-D for any domain D. So $\mathcal{MQ}$-D is $\mathcal{NP}$-hard.

$\mathcal{NP}$-completeness. Although [PG97] claims that $\mathcal{MQ}$ over *any* division ring is $\mathcal{NP}$-complete, we do not agree with this result. The reason is, that [PG97] omits to show that MQ-$D \in \mathcal{NP}$.

To show this, we consider the (slightly modified) algorithm from Section 2.5.1 for m equations in n variables over D:

1. Guess an assignment $A \in R^n$ for $(x_1, \ldots x_n)$
2. Check if all m equations are satisfied by A.
3. Output `true` or go to an infinite loop, respectively.

The crucial part is Step (2): Although this checking can be done in polynomial time over GF(2), this is not true in general: consider the rings $\mathbb{R}$ and $\mathbb{C}$. Neither has zero divisors, both have a multiplicative neutral, and therefore the $\mathcal{NP}$-hardness proof from above applies. However, addition and multiplication in the two structures $\mathbb{C}$ and $\mathbb{R}$ are not necessarily polynomial time operations and hence Step (2) may take more than $\mathcal{NP}$-time. So in general, MQ-D is $\mathcal{NP}$-hard. If all ring operations can be done in polynomial time, it is also $\mathcal{NP}$-complete.

Remark. The question whether MQ-$\mathbb{Z} \in \mathcal{NP}$ has not been answered in this section. The key question is the number of bits which are needed to encode a single assignment A. Closely related is the status of $\mathcal{MQ}$-$\mathbb{N}$. Although $\mathbb{N}$ is not a domain, it seems to be possible to transfer MQ-GF(2) to $\mathbb{N}$, using a similar algorithm as outlined in Section 2.5.2. However, as this thesis deals with using $\mathcal{MQ}$ for signature and encryption, and neither $\mathbb{Z}$ nor $\mathbb{N}$ seem to be of use for this aim, we will not investigate this question further.

2.5.3 Discussion

In this section, we showed that $\mathcal{MQ}$ over GF(2) and also MQ over some domains is $\mathcal{NP}$-complete. This is especially the case when these domains are finite (and hence are fields, as every finite domain is also a finite field). However, for general domains $\mathcal{MQ}$ is $\mathcal{NP}$-hard, but not necessarily $\mathcal{NP}$-complete.

In general, being $\mathcal{NP}$-complete does not imply that a public key crypto-system using this problem is automatically secure. A counterexample are crypto-systems using the knapsack problem. Although the knapsack problem is $\mathcal{NP}$-complete [GJ79, p. 65], most of these crypto-systems were broken, cf [MvOV96, Sec. 8.6] for an overview. However, for the $\mathcal{MQ}$-problem, there is strong empirical and theoretical evidence,*e.g.*, [CKPS00, CGMT02], that it is also hard on average (even with embedded trapdoor) and hence can be used as basis for a secure public key crypto system.

Still, for finite domains D, the corresponding problem $\mathcal{MQ} - D$ can be solved in nondeterministic polynomial time and hence, is $\mathcal{NP}$-complete.

2.6 Related Problems

As we saw in the previous section, the $\mathcal{MQ}$-problem is believed to be computationally hard as it is $\mathcal{NP}$-complete. Still, this is not sufficient for the construction of secure public key schemes. In this section, we outline two more problems which are used in the context of these schemes.

2.6.1 Isomorphism of Polynomials

For the construction of secure public key systems based on polynomial equations over finite fields, the security of the Isomorphism of Polynomials problem or IP-problem [Pat96b] is also important. With IP-problem we mean the difficulty to find affine transformations $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ such that $\mathcal{P} = T \circ \mathcal{P}' \circ S$ for given polynomial vectors $\mathcal{P}, \mathcal{P}'$. In particular, the private key in such systems is usually the triple $(S, \mathcal{P}', T)$, cf Section 2.3, and the public key the polynomial vector $\mathcal{P}$. Hence, if the IP-problem was easy, the security of these schemes would be jeopardised. Therefore, these constructions have to make the (often not explicitly stated) assumption that the corresponding IP-problem is difficult. If the central map $\mathcal{P}'$ has a special structure — which is the case for all systems based on the difficulty of solving a system of polynomial equations over a finite field — the corresponding IP-problem may become easy to solve and the system can be broken exploiting this weakness, cf *e.g.* [KS98, GC00, WBP04] for examples of such attacks.

A discussion of the security of the general IP-problem can be found in [Pat96b, PGC98b, GMS02]. In this context, the IP-problem with only one secret plays an important role. To fit in our framework, we assume here that T is given or equal to the identity transformation, hence the only unknown part is the affine transformation S. Interestingly, [LP03, Per05] shows that the IP-problem with one secret can be solved easily for if $m \geq n$, *i.e.*, the corresponding constructions are cryptographically insecure.

2.6.2 MinRank

When cryptanalysing $\mathcal{MQ}$-schemes, we sometimes face an instance of the so-called MinRank problem. Due to its importance, we introduce it formally here: let $(M_1, \ldots, M_k)$ be a sequence of $k \in \mathbb{N}$ matrices over $\mathbb{F}^{n \times n}$ each. Moreover, let $r \in \mathbb{N}$. For the MinRank-problem, we are interested in finding a linear combination of the above matrices, *i.e.*, a vector $\lambda \in \mathbb{F}^k$ such that

$$\text{Rank}\left(\sum_{i=1}^{k} \lambda_i M_i\right) \leq r\,.$$

Naturally, the above problem can be extended to other fields. However, when stated over finite fields, it is $\mathcal{NP}$-complete [BFS96].

In special cases, namely when the private key has the same structure as for STS-schemes, cf Section 3.1.2, the problem becomes tractable. In particular, [GC00] gives two algorithms with complexity $O(q^r)$, respectively, where $r \in \mathbb{N}$ is the number of new variables in an STS scheme. The question remains open if a more efficient algorithm for these cases or even the general MinRank-problem exists. A positive answer would have serious implications for the security of several schemes based on the $\mathcal{MQ}$-problem as the MinRank-problem has been used in the cryptanalysis of several systems, *e.g.*, in [CSV93, CSV97, KS98, KS99, GC00, WBP04].

Chapter 3

Constructions for $\mathcal{MQ}$-trapdoors

In the previous chapter, we have introduced the general $\mathcal{MQ}$-problem and have also explained how to embed a trapdoor into it. In this chapter, we consider special constructions of this trapdoor. We start with two examples which use only a finite field $\mathbb{F}$, denoted UOV (unbalanced oil and vinegar) and STS (stepwise triangular system). We then move on to the two schemes MIA (Matsumoto-Imai Scheme A) and HFE (hidden field equations); both use a ground field $\mathbb{F}$ and an extension field $\mathbb{E}$. Following this, we show different modifiers for $\mathcal{MQ}$-trapdoors.

Our own achievement in this chapter is the STS trapdoor which was not known in its full generality before [WBP04]. Moreover, we put all trapdoors known so far into the following taxonomy and hence, clarified the area. In addition, the modifiers sparse polynomials ("s"), vinegar variables ("v"), and internal perturbation ("i") have not been treated in their full generality before. In addition, the homogenising ("h") modifier has been developed during this thesis.

3.1 Basic Trapdoors

3.1.1 Unbalanced Oil and Vinegar Schemes: UOV

The "Unbalanced Oil and Vinegar" (UOV) scheme was introduced in [KPG99], cf [KPG03] for an extended version of this paper. UOV is a generalisation of the original Oil and Vinegar scheme of Patarin [Pat97].

Definition 3.1.1 *Let $\mathbb{F}$ be a finite field and $n, m \in \mathbb{N}$ with $m < n$ and coefficients $\alpha'_i, \beta'_{i,j}, \gamma'_{i,j,k} \in \mathbb{F}$. We say that the polynomials below are central equations* *in UOV-shape:*

$$p_i(x'_1, \ldots, x'_n) := \sum_{j=1}^{n-m} \sum_{k=1}^{n} \gamma'_{i,j,k}\, x'_j x'_k + \sum_{j=1}^{n} \beta'_{i,j}\, x'_j + \alpha'_i\,.$$

In this context, the variables x'_i for $1 \leq i \leq n-m$ are called the "vinegar" variables and x'_i for $n-m < i \leq n$ the "oil" variables. We also write $o := m$ for the number of oil variables and $v := n-m = n-o$ for the number of vinegar variables. Note that the vinegar variables are combined quadratically, while the oil variables are only combined with vinegar variables in a quadratic way. Therefore, assigning random values to the vinegar variables results in a system of linear equations in the oil variables which can then be solved, *e.g.*, using Gaussian elimination. The above notation clearly has some redundancies: knowing n and m, we can readily compute o and v — while v and m allow us to compute n and o. The problem in this context is that the papers about these schemes use very different notation. With the above settings, we use a kind of "generalised notation" which suits most of them.

Moreover, Unbalanced Oil and Vinegar schemes (UOV) omit the affine transformation T but only use $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$. To fit in our framework, we set it to be the identity transformation, *i.e.*, we have $T = id$ for UOV by definition. UOV is able to omit T as all equations have exactly the same shape. Hence, we do not need T to hide any special structure. Moreover, using the ideas of equivalent keys, cf Section 5, we can actually show that the transformation T could always be moved into the central equations $\mathcal{P}'$ and hence, does not give any gain in security.

The UOV scheme can only be used for signature schemes as we need $v \geq 2o$ for a secure construction; we give a more detailed security evaluation in Section 4.3 and only point out a few milestones here. The first attack against the original OV, *i.e.*, with parameters $o = v$ or $n = 2m$ can be found in [KS98]. This attack has been extended to UOV in [KPG99]. The latest security evaluation — also taking Gröbner bases into account, can be found in [BWP05]. As shown in all these papers, we have on average q^v different pre-images $x \in \mathbb{F}^n$ for a given vector $y \in \mathbb{F}^m$, so decryption is by no means efficient. In a nutshell, the most efficient attacks have a complexity of $O(q^{v-m-1}m^4) = O(q^{n-2m-1}m^4)$ and are due to [KPG99].

While being the easiest trapdoor shown in this thesis, it is surprisingly also the only safe basic trapdoor. We discuss results on the security of all trapdoors discussed in this thesis in Chapter 4 but want to point out that all basic trapdoors can be used to construct secure schemes when combined with the correct modifiers.

3.1.2 Stepwise Triangular Systems: STS

Another approach to obtain an invertible central map is used in step-wise triangular systems (STS), introduced in [WBP04]. As UOV, the class STS is defined over a finite field $\mathbb{F}$ and use a special structure for the central equations $\mathcal{P}'$ to allow easy inversion (cf Figure 3.1 for regular STS). Here, the step-width (number

$$
\begin{array}{ll}
\text{Step } 1 & \left\{ \begin{array}{ll} p'_1 & (x'_1, \dots, x'_r) \\ & \vdots \\ p'_r & (x'_1, \dots, x'_r) \end{array} \right. \\
\vdots & \\
\text{Step } l & \left\{ \begin{array}{ll} p'_{(l-1)r+1} & (x'_1, \dots, x'_r, \quad \dots, \quad x'_{(l-1)r+1}, \dots, x'_{lr}) \\ & \vdots \\ p'_{lr} & (x'_1, \dots, x'_r, \quad \dots, \quad x'_{(l-1)r+1}, \dots, x'_{lr}) \end{array} \right. \\
\vdots & \\
\text{Step } L & \left\{ \begin{array}{ll} p'_{(L-1)r+1} & (x'_1, \dots, x'_r, \quad \dots, \quad x'_{(l-1)r+1}, \dots, x'_{lr}, \quad \dots, \quad x'_{n-r+1}, \dots, x_n) \\ & \vdots \qquad\qquad\qquad \vdots \\ p'_{Lr} & (x'_1, \dots, x'_r, \quad \dots, \quad x'_{(l-1)r+1}, \dots, x'_{lr}, \quad \dots, \quad x'_{n-r+1}, \dots, x_n) \end{array} \right.
\end{array}
$$

Figure 3.1: Central Equations p'_i in a Regular STS Scheme

of new variables) and the step-height (number of new equations) is controlled by the parameter r. As usual, we use m for the number of equations and n for the number of variables. In addition, we denote by L the number of layers, q the size of the ground field $\mathbb{F}$, and r the step-width.

Let $r_1, \dots, r_L$ be L integers such that $r_1 + \cdots + r_L = n$, the number of variables, and $m_1, \dots, m_L \in \mathbb{N}$ such that $m_1 + \cdots + m_L = m$, the number of equations. Here r_l represents the number of new variables (step-width) and m_l the number of equations (step-height), both in step l for $1 \le l \le L$. In a general step-wise Triangular Scheme (gSTS), the m_l private quadratic polynomials of each layer l contain only the variables x'_k with $k \le \sum_{j=1}^{l} r_j$, *i.e.*, only the variables defined in all previous steps plus r_l new ones. The overall shape of the private polynomials leads to the name step-wise Triangular Scheme (STS), cf Figure 3.1 for regular STS. We want to stress in this context that we do not assume any specific structure for the private polynomials $p'_1, \dots, p'_m$ here. In particular, all coefficients $\gamma'_{i,j,k}, \beta'_{i,j}, \alpha'_i \in \mathbb{F}$ for these polynomials may be chosen at random.

When not mentioned otherwise, we concentrate on regular STS schemes (rSTS or STS for short) in this section to simplify explanations. For regular STS schemes we set $r_1 = \cdots = r_N = m_1 = \cdots = m_L$, which we denote by r. Consequently, we have $n = m = Lr$.

To invert a system of central equations $\mathcal{P}'(x') = y'$ for given $y' \in \mathbb{F}^m$, we exploit the step-structure: in each level l, we have q^r possible vectors and only need to keep the intermediate values $(x'_{(l-1)r+1}, \ldots, x'_{lr})$ which satisfy the corresponding equations

$$\begin{array}{rcl} y'_{(l-1)r+1} & = & p'_{(l-1)r+1}(x'_1, \ldots, x'_{lr}) \\ & \vdots & \\ y'_{lr} & = & p'_{lr}(x'_1, \ldots, x'_{lr}) \end{array}$$

for given $y'_{(l-1)r+1}, \ldots, y'_{lr} \in \mathbb{F}$. Having a bijective structure in each level makes sure we get only one solution — this way, STS becomes particularly efficient. However, we impose some conditions on the coefficients $\gamma'_{i,j,k}, \beta'_{i,j}, \alpha'_i \in \mathbb{F}$ this way. Anyway, in a signature scheme, it is even sufficient if we only get *one* solution for the corresponding equation. For general STS, we use the same idea but for each individual layer and hence with a different number of equations and variables. However, observe that the legitimate user has a workload growing with q^r which implies that this number cannot be too large if there is no special trapdoor embedded for each layer. Section 6.2 presents examples of such constructions with a special trapdoor embedded.

After outlining both regular and general step-wise triangular schemes, we give a brief account of constructions suggested so far. We begin with the Birational Permutation Schemes of Shamir [Sha93]. They are regular STS schemes with $r = 1$. However, as previously mentioned, they are not defined over a (small) finite field but over a (large) finite ring. So strictly speaking, they are not STS schemes although they are clearly related. In contrast, the TPM (Triangle Plus Minus, [GC00]) class of Goubin and Courtois coincides with STS for the parameters $r_1 = u$, $m_L = v$, $m_1 = \cdots = m_{L-1} = r_2 = \cdots = r_L = 1$, *i.e.*, we remove $u \in \mathbb{N}$ initial layers, add $v \in \mathbb{N}$ polynomials in the last step, and have exactly one new variable at all intermediate levels. TPM is a subclass of STS as it is not defined over a ring but over a field, and hence, is an example of an $\mathcal{MQ}$-scheme.

Shamir's scheme was broken shortly after its publication in [CSV93, The95, CSV97]. The TPM scheme of Goubin and Courtois has been broken in the same paper that proposed it [GC00]. In fact, the aim of their construction was to show that Moh's TTM (Moh's Tame Transformation Method, [Moh99]) construction is weak.

The schemes RSE(2)PKC and RSSE(2)PKC, proposed by Kasahara and Sakai [KS04c, KS04b], also fall in the class of STS schemes. The definition of these schemes can be found in Section 4.4.5. Both schemes — and actually the whole

STS class — have been broken in [WBP04]. We will see an overview of these attacks in Section 4.4.

3.1.3 Matsumoto-Imai Scheme A: MIA

The scheme MIA is due to Matsumoto and Imai [IM85, MI88]. It is the first scheme in this thesis which uses two different finite fields, namely a ground field $\mathbb{F}$ and an extension field $\mathbb{E}$. As outlined in Section 2.1, we use the canonical bijection $\phi : \mathbb{E} \to \mathbb{F}^n$ to transfer elements between the extension field $\mathbb{E}$ and the vector space $\mathbb{F}^n$. This way we have all tools at hand for the following definition.

Definition 3.1.2 *Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements, $\mathbb{E}$ be its n-th degree extension, and $\phi : \mathbb{E} \to \mathbb{F}^n$ the canonical bijection between this extension field and the corresponding vector space (cf Definition 2.1.6). In addition, let $\lambda \in \mathbb{N}$ be an integer such that $\gcd(q^n - 1, q^\lambda + 1) = 1$. We then say that the following central equation over the extension field $\mathbb{E}$ is of MIA-shape:*

$$P'(X') := (X')^{q^\lambda+1} \text{ with } X' \in \mathbb{E}\,.$$

We also write $\mathcal{P}' := \phi \circ P' \circ \phi^{-1}$ to obtain a system of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials. The corresponding algorithm can be found in Lemma 2.4.1.

The restriction $\gcd(q^n - 1, q^\lambda + 1) = 1$ is necessary first to obtain a permutation polynomial and second to allow efficient inversion of $P'(X')$. Indeed, the equation $h.(q^\lambda + 1) \equiv 1 \pmod{q^n - 1}$ has exactly one solution $h \in \mathbb{N}$ with $h < q^n - 1$, as we have the previously mentioned gcd-condition on λ. Given h, we can solve $Y' = P'(X')$ as $(Y')^h = X'^{[h.(q^\lambda+1)]} = X'$ by raising Y' to the power of h. Note that these operations take place in the n-th dimensional extension $\mathbb{E}$ of the finite field $\mathbb{F}$. All in all, this approach is similar to RSA. However, the hardness of MIA is not based on the difficulty of finding the exponent h but in the intractability to obtain transformations S, T for given polynomial equations $\mathcal{P}, \mathcal{P}'$ (IP-problem, cf Section 2.6.1). As we saw in Section 2.4, the monomial $(X')^{q^{\lambda+1}}$ can be expressed in terms of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations and hence be used as a trapdoor for an $\mathcal{MQ}$-problem. More on this topic can be found in Lemma 2.4.1 and also Theorem 2.4.9.

Note that MIA is insecure, due to a very efficient attack by Patarin [Pat95]. Moreover, we want to point out that Geiselmann *et al.* showed how to reveal the constant parts of these transformations [GSB01]. Hence, having S, T affine instead of linear does not seem to enhance the overall security of MIA. The papers [WP05b, WP05c] discuss the question of equivalent keys for MIA and some variations. Their ideas are summarised in Section 5.

Remark 3.1.3 *In the paper [MI88], MIA was introduced under the name C^*. Moreover, it used the branching modifier (cf Section 3.2.4) by default. As branching has been attacked very successfully, C^* has been used without this modification for any later construction,* e.g., *[CGP00c, CGP02, CGP00a, CGP03a]. However, without the branching condition, the scheme C^* coincides with the previously suggested "Scheme A" from [IM85]. To acknowledge this historical development, we decided to use the earlier notation and call the scheme presented in this section "MIA" for "Matsumoto-Imai Scheme A". As an additional benefit, the notation becomes more uniform as all basic schemes are now named with three-letter acronyms.*

3.1.4 Hidden Field Equations: HFE

After breaking MIA, Patarin generalised the underlying trapdoor to "Hidden Field Equations" [Pat96b]. This generalisation aims at the central equations and uses a univariate *polynomial* rather than a univariate *monomial* here. But the basic idea of MIA, *i.e.*, to mix a given ground field with one of its extension fields is still used in HFE as we see in the following

Definition 3.1.4 *Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements, $\mathbb{E}$ be its n-th degree extension, and $\phi : \mathbb{E} \to \mathbb{F}^n$ the canonical bijection between this extension field and the corresponding vector space (cf Definition 2.1.6). Moreover, let $P(X)$ a univariate polynomial over $\mathbb{E}$ with*

$$P'(X') := \sum_{\substack{0 \le i,j \le d \\ q^i+q^j \le d}} C'_{i,j} X'^{q^i+q^j} + \sum_{\substack{0 \le k \le d \\ q^k \le d}} B'_k X'^{q^k} + A'$$

$$\text{where} \begin{cases} C'_{i,j} X^{q^i+q^j} & \text{for } C'_{i,j} \in \mathbb{E} \text{ are the quadratic terms,} \\ B'_k X^{q^k} & \text{for } B'_k \in \mathbb{E} \text{ are the linear terms, and} \\ A' & \text{for } A' \in \mathbb{E} \text{ is the constant term} \end{cases}$$

for $i, j \in \mathbb{N}$ and a degree $d \in \mathbb{N}$. Now we say the central equations $\mathcal{P}' := \phi \circ P' \circ \phi^{-1}$ are in HFE-shape.

As the degree of the polynomial P' is bounded by d, this allows efficient inversion of the equation $P'(X') = Y'$ for given $Y' \in \mathbb{E}$ and small d. An overview of possible algorithms for this problem can be found in [Pat96b, Section 5]; in a nutshell, these algorithms depend both on the size of the dimension n of the extension field $\mathbb{E}$ and the degree d of the central polynomial P. Hence, from an efficiency point of view, both should be rather small. Moreover, in contrast to MIA, HFE is in general no surjection. Possible ways to overcome this problem are outlined in Section 2.3.3.

As for MIA, we notice that the HFE polynomial $P'(X')$ can be expressed as $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations $\mathcal{MQ}(\mathbb{F}^n)$. They are therefore a candidate for central equations $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n)$. For more details on how to express these polynomials over the vector space $\mathbb{F}^m$, see Section 2.4 and in particular Lemma 2.4.5 and Theorem 2.4.9.

From a cryptanalytic point of view, the basic HFE scheme is broken: an efficient key recovery attack, using the MinRank-problem (cf Section 2.6.2), has been demonstrated in [KS99]. An inversion attack which uses both Gröbner bases and general linearization methods has been shown in [FJ03]. In [SG03] we find an attack which works better if n is not a prime, *i.e.*, we have splitting fields. A more detailed discussion of HFE can be found in [Pat96b, Cou01, WP04]. Here, [Pat96b] gives some general considerations of HFE after its development, *e.g.*, a general linearization attack against all multivariate schemes (cf Section 4.2), while [Cou01] summarised the situation of HFE in 2001 and also improves over the attack from [KS99]. The latest such summary of attacks can be found in [WP04]. In particular, this paper outlines two versions of HFE which are secure against all known attacks. Finally, [WP05b, WP05c] show that HFE allow many equivalent keys and hence, waste memory. The corresponding ideas can be found in Chapter 5.

3.1.5 Taxonomy and Discussion

The four trapdoors discussed above are all basic trapdoors known so far. We notice that all of them are rather old: the first were MIA (1985) and STS (1993 in Shamir's birational permutations and 2004 in STS), followed by HFE (1996) and UOV (1997 as OV and 1999 as UOV). Apart from UOV with well-chosen parameters, all basic trapdoors have to be considered broken. Unfortunately, UOV is rather inefficient in terms of signature expansion as it has a rate of 3 between m, and n, *i.e.*, it has an overall signature expansion rate of 3. Moreover, UOV can not be used to construct a secure encryption scheme. Therefore, we discuss some generic modifications in the next section. A nice property of these modifiers is that they can be used in combination with any of these basic trapdoors (see below). Moreover, we will see how it is possible to combine several basic trapdoors to more elaborated $\mathcal{MQ}$-systems in Section 6.2.

Before doing so, we build up a taxonomy to get a better view on the different trapdoors used so far. A graphical representation of this idea is given in Figure 3.2. Using the finite fields as a first criterion, we see that MIA and HFE form the class of "mixed field" schemes: both use the ground field $\mathbb{F}$ and an extension field $\mathbb{E}$ to construct a trapdoor. Therefore, both are vulnerable to attacks using Gröbner bases as these can exploit the structure of the extension field and the rather low number of univariate monomials when compared to a random system

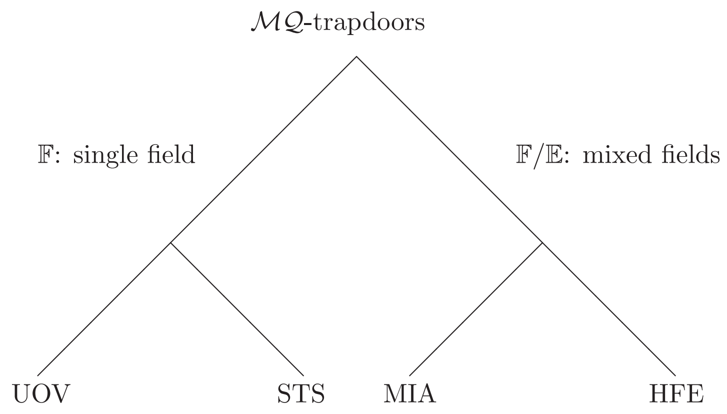

Figure 3.2: Taxonomy of the Basic $\mathcal{MQ}$-trapdoors

of equations. The same is true for the linearization attack as discussed,*e.g.*, in [JKJMR05]. In contrast, UOV and STS are "single field" systems as they only use the ground field $\mathbb{F}$ but they construct their trapdoor using special conditions for the polynomials $p'_1, \ldots, p'_m$: the concept of vinegar variables for UOV and a layer- or step-structure for STS. In both cases, the ranks of these central equations proved to be a serious vulnerability. While it was possible for UOV to fix this problem with well-chosen parameters, STS does not allow such an option.

At first glance, MIA is a subclass of the HFE system: while MIA uses only one monomial, HFE uses a whole polynomial. So from a cryptanalytic point of view, HFE is much stronger than MIA and all attacks which break HFE will also defeat MIA. The converse is not true though. Moreover, if we inspect both schemes more closely, we see differences: MIA uses a monomial of a high degree, while HFE relies on the existence of efficient root finding algorithms for polynomials — and therefore needs a much smaller degree d than MIA. Hence, using implementation as a criterion, we kept both schemes in different classes.

3.2 Generic Modification on $\mathcal{MQ}$-schemes

As we saw in the previous section, most basic trapdoors are insecure. Fortunately, we do not only have these four basic trapdoors, but also several generic "modifiers". Hence, to construct secure schemes, we can make use of these "modifications" of the basic building blocks. As we will see below, these modifications are quite generic as we can apply them (at least in theory) to *any* of the above trapdoors. However, for some schemes, there are modifications which prove more efficient.

3.2.1 Minus method: "-"

Although this modification looks rather easy, it proves powerful to defeat a wide class of cryptographic attacks against several $\mathcal{MQ}$-schemes, including Gröbner bases and linearization attacks. The minus method has been introduced in [Sha93]. In this new construction, we set $\tilde{m} := m - r$ for some $r \in \mathbb{N}$ and define the public key equations as $\mathcal{P} := R \circ T \circ \mathcal{P}' \circ S$. In this context, the function $R : \mathbb{F}^m \to \mathbb{F}^{\tilde{m}}$ denotes a *reduction* or *projection*. Details on this function are given in Section 2.4. In addition, we have the affine transformations $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n), T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ and the private system of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$. Less loosely speaking, we consider the function $R(y_1, \ldots, y_m) := (y_1, \ldots, y_{m-r})$, *i.e.*, we neglect the last r components of the output vector $(y_1, \ldots, y_n)$. As a consequence, a given public key $\hat{\mathcal{P}} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ is transfered to a new key $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^{\tilde{m}})$, cf Figure 3.3.

$$
\begin{array}{rcl}
\hat{p}_1(x_1, \ldots, x_n) & \rightarrow & p_1(x_1, \ldots, x_n) \\
 & \vdots & \\
\hat{p}_{m-r}(x_1, \ldots, x_n) & \rightarrow & p_{m-r}(x_1, \ldots, x_n) \\
 & & \\
\left.\begin{array}{c} \hat{p}_{m-r+1}(x_1, \ldots, x_n) \\ \vdots \\ \hat{p}_m(x_1, \ldots, x_n) \end{array}\right\} & \textit{discarded} &
\end{array}
$$

Figure 3.3: Minus modification for $\hat{\mathcal{P}}$ being transformed to $\mathcal{P}$

For MIA (or C^*), the corresponding minus variation is called MIA- (or C^{*--}) and has been discussed in [PGC98a]. For HFE, we derive HFE-. In particular, the attacks from [KS99, FJ03] are no longer effective against this variation.

3.2.2 Plus method: "+"

As the name suggests, the plus method adds equations to the public key rather than removing them. To the knowledge of the author, this method has been first discussed in [Pat96b, PGC98a]. In a nutshell, the legitimate user inserts a total of $a \in \mathbb{N}$ random quadratic equations $(\pi_1, \ldots, \pi_a) \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^a)$ without a trapdoor to the central equations. Let $\tilde{\mathcal{P}} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^{\tilde{m}})$ be the initial central equations and $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ be the new central equations. We have $m := \tilde{m} + a$ for $m, \tilde{m} \in \mathbb{N}$ and

$$\begin{array}{rcl}
p'_1(x'_1, \ldots, x'_n) & := & \tilde{p}_1(x'_1, \ldots, x'_n) \\
& \vdots & \\
p'_{\tilde{m}}(x'_1, \ldots, x'_n) & := & \tilde{p}_{\tilde{m}}(x'_1, \ldots, x'_n) \\
p'_{\tilde{m}+1}(x'_1, \ldots, x'_n) & := & \pi_1(x'_1, \ldots, x'_n) \\
& \vdots & \\
p'_m(x'_1, \ldots, x'_n) & := & \pi_a(x'_1, \ldots, x'_n)
\end{array}$$

Following the notation earlier introduced in this thesis, we see that the polynomials $p'_1, \ldots, p'_m$ are components of the (new) central equations $\mathcal{P}'$ and $\tilde{p}_1, \ldots, \tilde{p}_{\tilde{m}}$ are components of the (old) central polynomial vector $\tilde{\mathcal{P}}$.

Initially, the plus method was suggested with three affine transformations $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n), T \in \mathrm{Aff}^{-1}(\mathbb{F}^{m'})$ and $U \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ rather than two transformations $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n), W \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ as described in this thesis. However, as proven in [Wol02a, Section 4.6], the two methods have equal security as the method with three affine transformations can always be expressed with two transformations and vice versa. We give an adapted version of this proof here. However, as $T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ is an affine transformation, it will affect the polynomials $\tilde{p}_1, \ldots, \tilde{p}_n$ in an affine way. We show that it is not necessary to mix $\tilde{p}_1, \ldots, \tilde{p}_n$ and $\pi_1, \ldots, \pi_a$ using a new affine transformation U but that it is sufficient to mix them with an affine transformation $W \in \mathrm{Aff}^{-1}(\mathbb{F}^{n+a})$ and to replace both affine transformations T and U by W. Figure 3.4 gives a graphical representation of both ideas. Here $p'_1, \ldots, p'_{n+a}$ denotes the overall result, $\tilde{p}_1, \ldots, \tilde{p}_n$ are the polynomials of the original central equations $\tilde{\mathcal{P}}$, and $\overline{p_1}, \ldots, \overline{p_n}$ are the intermediate result after applying the affine transformation T.

To show that these two different ways of incorporating the random polynomials $\pi_1, \ldots, \pi_a$ are equivalent, we will study how the two affine transformations T

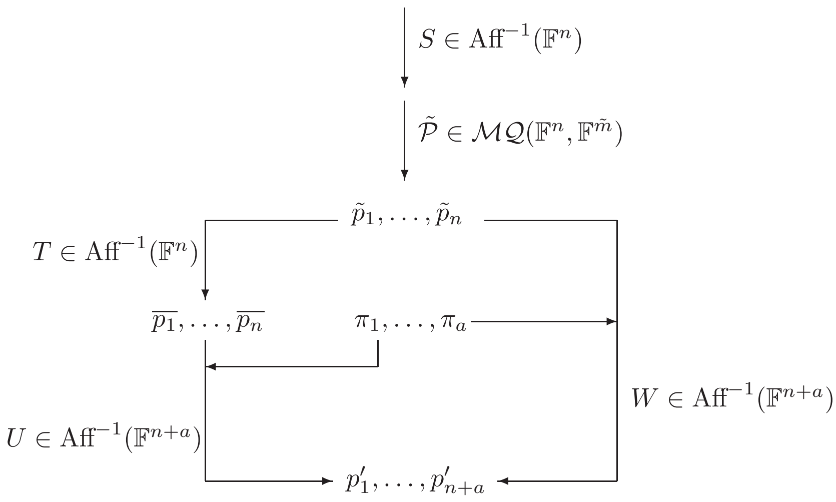

Figure 3.4: $\mathcal{MQ}$-trapdoor with three (left) and two (right) affine transformations

and U affect the different polynomial vectors involved. Before we start, we express the affine transformation U as one matrix $M_U \in \mathbb{F}^{n+a \times n+a}$ and a vector $v_u \in \mathbb{F}^{n+a}$. The affine transformation T will be expressed in a non-standard way. Rather than having one matrix $M_T \in \mathbb{F}^{n \times n}$ and one vector $v_t \in \mathbb{F}^n$, we will use a matrix $\overline{M_T} \in \mathbb{F}^{(n+a)\times(n+a)}$ and one vector $\overline{v_t} \in \mathbb{F}^{n+a}$. Using the coefficients $(m_t)_{i,j}$ of the matrix M_T and the coefficients $(v_t)_i$ of vector v_t, they are coefficient-wise defined as follows:

$$\begin{aligned}
(\overline{m_t})_{i,j} &:= \begin{cases} (m_t)_{i,j} & \text{, for } i,j \le n \\ 1 & \text{, for } i,j > n \text{ and } i = j \\ 0 & \text{, otherwise} \end{cases} \\
(\overline{v_t})_i &:= \begin{cases} (v_t)_i & \text{, for } i \le n \\ 0 & \text{, otherwise} \end{cases}
\end{aligned}$$

So in terms of matrix multiplication and vector addition, we can express the last two steps of $\mathcal{MQ}+$ as

$$M_U \left[\overline{M_T} \begin{pmatrix} \tilde{p_1} \\ \vdots \\ \tilde{p_n} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \overline{v_t} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \pi_1 \\ \vdots \\ \pi_a \end{pmatrix} \right] + v_u = M_U \left[\overline{M_T} \begin{pmatrix} \tilde{p_1} \\ \vdots \\ \tilde{p_n} \\ \pi_1 \\ \vdots \\ \pi_a \end{pmatrix} + \overline{v_t} \right] + v_u$$

$$= (M_U \overline{M_T}) \begin{pmatrix} \tilde{p_1} \\ \vdots \\ \tilde{p_n} \\ \pi_1 \\ \vdots \\ \pi_a \end{pmatrix} + (M_U \overline{v_t} + v_u) = M_W \begin{pmatrix} \tilde{p_1} \\ \vdots \\ \tilde{p_n} \\ \pi_1 \\ \vdots \\ \pi_a \end{pmatrix} + v_w$$

for some $M_W \in \mathbb{F}^{(n+a)\times(n+a)}, v_w \in \mathbb{F}^{n+a}$. Moreover, both matrices $\overline{M_T}$ and M_U are invertible, hence $M_W = M_U \overline{M_T}$ is also invertible. As all matrices are invertible, we can always compute one affine transformation for any two other given transformations. So from a cryptographic point of view, we can apply W directly to $(\tilde{p_1}, \ldots, \tilde{p_n}, \pi_1, \ldots, \pi_a)^t$ rather than working with the two transformations T, U. We show this with the following argument: keep the transformation T fixed, and choose transformation U at random. As $\overline{M_T}$ is invertible, two different transformations U, U' will yield two different transformations W, W', so their number is the same. In terms of probability, each transformation W has the same probability to appear for T fixed and U chosen at random. This is also true when we allow different values for the transformation T: for each T, there is one (and exactly one) U which yields a specific W. So the probability for a specific W to appear does not change by allowing T to have different values. So rather than choosing T and U at random and then compute W, we can choose W at random without changing the probability for any specific $W \in \mathrm{Aff}^{-1}(\mathbb{F}^{n+a})$ to appear.

When it was proposed, the plus method was thought to enhance the security of schemes like MIA or HFE. However, a more detailed cryptanalysis showed that this is not the case. In addition, signature schemes have a workload increasing with q^a as only q^{-a} of all solutions to the original problem $\tilde{\mathcal{P}}$ are also a solution for the a equations (without trapdoor) $\pi_1, \ldots, \pi_a$. Hence, this method has not received much attention lately.

3.2.3 Subfield method: "/"

A big drawback of public key schemes based on the $\mathcal{MQ}$-problem are their rather large public keys. To overcome this problem we can choose all their coefficients in the transformations $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ and also the central equations $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ in a proper subfield $\tilde{\mathbb{F}}$ of the ground field $\mathbb{F}$. This way, the size of both the public and the private key decrease by a factor of $\log_2 \tilde{\mathbb{F}} / \log_2 \mathbb{F}$. For example, choosing $\mathbb{F} = \mathrm{GF}(256)$ and $\tilde{\mathbb{F}} = \mathrm{GF}(2)$, we reduce the size of all keys with a factor of 8. The method works as subfields are closed under addition and multiplication and hence, choosing all coefficients in the components $S, \mathcal{P}', T$ of the private key in a proper subfield $\tilde{\mathbb{F}}$ ensures that the public key $\mathcal{P} := T \circ \mathcal{P}' \circ S$ has all its coefficients in $\tilde{\mathbb{F}}$ rather than $\mathbb{F}$. Hence, the space for storing these coefficients drops from $\log_2 q$ to $\log_2 |\tilde{\mathbb{F}}|$. On the other hand, the message space $\mathbb{F}^n$ is not affected by this change as all operations are still defined over the initial ground field $\mathbb{F}$.

This method was introduced in [Pat96b] and has been used in the first version of the Sflash signature scheme [CGP00c] as submitted to the NESSIE project [NES]. In addition, it has been used in the context of UOV [KPG03]. In both cases, the construction has been shown to be insecure [GM02, BWP05]. Similar conclusions for HFE have been drawn in [SG03]. All in all, we strongly discourage the use of this "subfield-trick" as it usually allows an easier cryptanalysis.

3.2.4 Branching: "⊥"

The idea of this modification is rather old and can already be found in [MI88]. A graphical representation using two branches with $n = n_1 + n_2$ and $m = m_1 + m_2$ for some $n_1, n_2, m_1, m_2 \in \mathbb{N}$ is given in Figure 3.5. For example in MIA, this

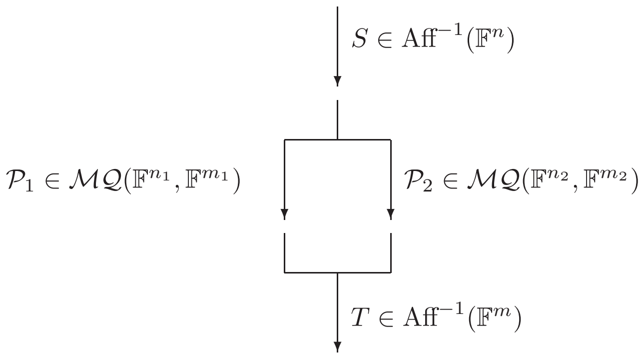

Figure 3.5: $\mathcal{MQ}$-trapdoor with two branches $\mathcal{P}_1, \mathcal{P}_2$

modification gives a speed-up for decryption, as we can reduce the dimension of the extension field $\mathbb{E}$ from n to n_1 and n_2. Hence, we are no longer confronted with a workload growing in $O(n^k)$ for some fixed $k \in \mathbb{R}$, $k > 1$, but only in $O(n_1^k + n_2^k)$ for the two smaller numbers n_1 and n_2. Similar conclusions can be drawn for all other basic trapdoors.

More general, the overall computational effort is reduced by partitioning both the polynomials $p'_1, \ldots, p'_m$ and the variables $x'_1, \ldots, x'_n$ in $B \in \mathbb{N}$ sets. Here, we call B the *branching number*. All computations for the central equations $\mathcal{P}'$ are then independently performed in these B sets. Note that we had $B = 2$ in the example of Figure 3.5. Formalising this idea, we decompose the number of variables into a B-dimensional vector over $\mathbb{N}$ such that $n = n_1 + \cdots + n_B$. Similar, we decompose the number of equations into $m_1, \ldots, m_B \in \mathbb{N}$ such that $m = m_1 + \cdots + m_B$. We use this notation to write down the branching structure, cf. Figure 3.6. At first glance this closely resembles the idea from STS, cf

$$
\begin{array}{ll}
\text{Branche } 1 & \left\{ \begin{array}{rl} p'_1 & (x'_1, \ldots, x'_{n_1}) \\ & \\ p'_{m_1} & (x'_1, \ldots, x'_{n_1}) \end{array} \right. \qquad \text{with } x'_i \in \mathbb{F} \\
\vdots & \\
\text{Branche } b & \left\{ \begin{array}{rl} p'_{m_1+\cdots+m_{b-1}+1} & (x'_{n_1+\cdots+n_{b-1}+1}, \ldots, x'_{n_1+\cdots+n_b}) \\ & \vdots \\ p'_{m_1+\cdots+m_b} & (x'_{n_1+\cdots+n_{b-1}+1}, \ldots, x'_{n_1+\cdots+n_b}) \end{array} \right. \\
\vdots & \\
\text{Branche } B & \left\{ \begin{array}{rl} p'_{m-m_b+1} & (x'_{n-n_B+1}, \ldots, x'_n) \\ & \\ p'_m & (x'_{n-n_B+1}, \ldots, x'_n) \end{array} \right.
\end{array}
$$

Figure 3.6: Central Polynomials p'_i with B branches

Section 3.1.2. However, there is an important difference here: while STS uses the variables from the previous layers (or "branches" for the "$\perp$" modification), this is not the case for the "$\perp$" modification. Here, all branches are completely independent from each other. Hence, all computations can be done in parallel, *e.g.*, in hardware, which allows a considerable speed-up. This was also the initial reason for proposing branching: having a more efficient public key scheme. Unfortunately, the articles [Pat95, Pat96a] give an algorithm for separating these branches. To the knowledge of the author, the most efficient algorithm for this problem has been given in [Fel01, Fel04]. It has an overall running time of $O(n^6)$ and is hence independent of the number or size of branches.

Therefore, we strongly discourage the use of the "$\perp$" modification in multivariate systems — though they lead to more efficient schemes. But this gain in efficiency is paid with a too high price on the security side.

3.2.5 Fixing: "f"

A similar idea to the minus "-" modification is the fixing "f" modification: instead of deleting some public key equations, we reduce the number of variables by explicitly assigning values to the variables $x_{n-f+1}, \ldots, x_n$ for a given parameter $f \in \mathbb{N}$. More formally, we pick a random vector $(a_1, \ldots, a_f) \in \mathbb{F}^f$ and partly evaluate the public key polynomials $p_1, \ldots, p_m$. This way, we obtain new polynomials $\tilde{p}_1, \ldots, \tilde{p}_m$ which now depend on the input variables $x_1, \ldots, x_{\tilde{n}}$ with

$$\boxed{\begin{array}{rcl} \tilde{p}_1(x_1, \ldots, x_{\tilde{n}}) & := & p_1(x_1, \ldots, x_{\tilde{n}}, a_1, \ldots, a_f) \\ & \vdots & \\ \tilde{p}_m(x_1, \ldots, x_{\tilde{n}}) & := & p_m(x_1, \ldots, x_{\tilde{n}}, a_1, \ldots, a_f) \end{array}}$$

Figure 3.7: Fixing Modification for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems $\tilde{\mathcal{P}}$ and $\mathcal{P}$

$\tilde{n} := n - f$ instead of $x_1, \ldots, x_n$. In Figure 3.7 we can see this idea explained with an old public key $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$, a new public key $\tilde{\mathcal{P}} \in \mathcal{MQ}(\mathbb{F}^{\tilde{n}}, \mathbb{F}^m)$ and a fixing vector $a \in \mathbb{F}^f$.

If the initial system did not have any linear or constant terms, *i.e.*, we set the coefficients $\beta_{i,j}$ and α_i equal to zero for $1 \leq i \leq m$ and $1 \leq j \leq n$, we can use the zero vector $(0, \ldots, 0) \in \mathbb{F}^f$ for fixing the variables $x_{n-f+1}, \ldots, x_n$, *i.e.*, we have $a = 0$ in the above setting. This way, we do not introduce new linear or constant terms and hence save public key space. From a cryptographic point of view, this does not introduce a weakness *if* the original idea of fixing is secure in the first place. Another way of looking at fixing is the use of the "inverse reduction" function from Lemma 2.4.7, *i.e.*, $R^{-1} : \mathbb{F}^n \to \mathbb{F}^{\tilde{n}}$ which is defined as

$$R^{-1}(x_1, \ldots, x_{\tilde{n}}) := (x_1, \ldots, x_{\tilde{n}}, 0, \ldots, 0)\,.$$

Now we can express the new public key as $\tilde{\mathcal{P}} := \mathcal{P} \circ R^{-1}$. This way we established the similarity between the minus and the fixing modification: in both cases, we reduce the structure of the public key by removing some information.

All in all, the idea works quite well with encryption schemes but gives a slowdown of q^f for signature schemes: we only have a probability of q^{-f} for a signature to have the correct values for $(x_{n-f+1}, \ldots, x_n) = (a_1, \ldots, a_f)$.

After being suggested in [Cou01], there has not been much work done on the security of this modification. In particular, it is unknown how the running time Gröbner attacks depends on this parameter f for systems such as MIA and HFE. Therefore, we suggest a deeper study of the "f" modification in connection with these basic trapdoors before using this modification.

3.2.6 Sparse Polynomials: "s"

In this section, we introduce the idea of using very sparse polynomials for the central map $\mathcal{P}'$. In particular, this means that all known attacks against these schemes have to be taken into account very carefully as the newly constructed polynomials only offer "on-the-edge" security. This idea has been used both by Yang and Chang [YC04a] and also by Wang, Hu, Lai, Chou, and Yang [WC04, WHL$^+$05] to construct fast asymmetric schemes.

Obviously, there is a clear benefit: instead of evaluating a total of $\tau(n)$ terms for each hidden polynomial p'_i with $1 \le i \le m$, we can concentrate on far less terms. This saves both time and memory. In particular, inverting these systems is now more time efficient.

However, the idea is rather new and there is not much known yet about hidden vulnerabilities of these schemes. Therefore, we suggest to study them in more depth before applying it to concrete schemes.

3.2.7 Vinegar Variables: "v"

The following modification has been introduced in the context of HFE by Kipnis, Patarin, and Goubin in [KPG99] for the HFE scheme. They called their new scheme HFEv for "Hidden Field Equations with vinegar variables" and use a different form for the central equations $\mathcal{P}'$. The basic idea is to hide the structure of the original central equations $\mathcal{P}'$ by multiplying the linear and constant terms with degree one and degree two terms, respectively. The overall public key polynomials $\mathcal{P}$ are still of degree two, and hence within the class of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials.

To the knowledge of the author, we are the first to present the "v" modification in a general form so it can be used with any trapdoor. In particular, the multivariate version of vinegar (cf Definition 3.2.2) has not been presented before.

Definition 3.2.1 *Let $\mathbb{E}$ be a finite field with degree $n' \in \mathbb{N}$ over its ground field $\mathbb{F}$, $v \in \mathbb{N}$ the number of vinegar variables, $n := n' + v$ the number of in-*

put variables, and P'(X') a polynomial over $\mathbb{E}$. Moreover, let $(z'_1, \ldots, z'_v) := s_{n-v+1}(x_1, \ldots, x_n), \ldots, s_n(x_1, \ldots, x_n)$ for s_i polynomials of $S(x)$ in multivariate representation, cf Definition 2.2.4. Then define the central polynomial

$$P'_{z'_1,\ldots,z'_v}(X') \;:=\; \sum_{0\le i,j<n} C_{i,j}X'^{q^i+q^j} + \sum_{k=0}^{n'-1} B_k(z'_1,\ldots,z'_v)X'^{q^k} + A(z'_1,\ldots,z'_v)$$

$$\text{where}\begin{cases} C_{i,j}X'^{q^i+q^j} & \text{for } C_{i,j}\in\mathbb{E} \text{ the quadratic terms,}\\ B_k(z'_1,\ldots,z'_v)X'^{q^k} & \text{for } B_k(z'_1,\ldots,z'_v) \text{ depending linearly on } z'_1,\ldots,z'_v \text{ and}\\ A(z'_1,\ldots,z'_v) & \text{for } A(z'_1,\ldots,z'_v) \text{ depending quadratically on } z'_1,\ldots,z'_v \end{cases}$$

Then we say the polynomial $P'_{z'_1,\ldots,z'_v}(X')$ is in univariate vinegar *shape.*

The condition that the $B_k(z_1, \ldots, z_v)$ are affine functions (*i.e.*, of degree 1 in the z_i at most) and $A(z_1, \ldots, z_v)$ is a quadratic function over $\mathbb{F}$ ensures that the public key as a whole is still quadratic over $\mathbb{F}$. In addition, we can obtain a similar definition for the case of multivariate quadratic polynomials:

Definition 3.2.2 *Let $\mathbb{F}$ be a finite field $\mathbb{F}$, $v \in \mathbb{N}$ the number of vinegar variables, and $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^{\tilde{n}}, \mathbb{F}^m)$ a polynomial-vector over $\mathbb{F}$ in $\tilde{n} \in \mathbb{N}$ input variables and with $m \in \mathbb{N}$ equations. Moreover, we consider the polynomial-vector $(z'_1, \ldots, z'_v) := s_{n-v+1}(x_1, \ldots, x_n), \ldots, s_n(x_1, \ldots, x_n)$ for s_i polynomials of $S(x)$ in multivariate representation, cf Definition 2.2.4. In addition we have $n := \tilde{n}+v$*

for the number of variables. Then define the central polynomials as

$$\begin{aligned}
p'_1(x_1,\ldots,x_{\tilde{n}}) &:= \sum_{1\le j\le k\le \tilde{n}} \gamma'_{1,j,k}x_jx_k \\
&\quad+\sum_{j=1}^{\tilde{n}} \beta'_{1,j}(z'_1,\ldots,z'_v)x_j + \alpha'_1(z'_1,\ldots,z'_v) \\
&\;\vdots \\
p'_i(x_1,\ldots,x_{\tilde{n}}) &:= \sum_{1\le j\le k\le \tilde{n}} \gamma'_{i,j,k}x_jx_k \\
&\quad+\sum_{j=1}^{\tilde{n}} \beta'_{i,j}(z'_1,\ldots,z'_v)x_j + \alpha'_i(z'_1,\ldots,z'_v) \\
&\;\vdots \\
p'_m(x_1,\ldots,x_{\tilde{n}}) &:= \sum_{1\le j\le k\le \tilde{n}} \gamma'_{m,j,k}x_jx_k \\
&\quad+\sum_{j=1}^{\tilde{n}} \beta'_{m,j}(z'_1,\ldots,z'_v)x_j + \alpha'_m(z'_1,\ldots,z'_v)
\end{aligned}$$

Here we have a new system of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations with only v input variables $A' \in \mathcal{MQ}(\mathbb{F}^v,\mathbb{F}^m)$ for $A' =: (\alpha'_1,\ldots,\alpha'_m)$ and a polynomial vector $B' \in Aff_0(\mathbb{F}^v,\mathbb{F}^{m\tilde{n}})$ with coefficients $B' =: (\beta'_{1,1},\beta'_{1,2},\ldots,\beta'_{m,\tilde{n}})$. Then we say the central polynomial $\mathcal{P}'$ is in multivariate vinegar *shape.*

We want to point out that the definition of the central equations $\mathcal{P}' = (p'_1,\ldots,p'_m)$ is the same as given in Section 2.2, but with a slight twist on the coefficients used: the linear coefficients β are replaced by non-homogeneous degree 1 polynomials, while the constant coefficients α are replaced by non-homogeneous degree 2 polynomials.

Inverting the central equation $P'(X') = Y'$ or $\mathcal{P}'(x') = y'$ for $X',Y' \in \mathbb{E}$ and $x,y \in \mathbb{F}^n$ requires to invert the original trapdoor q^v times. For a signature scheme, this is not a problem as finding a solution for any of these equations will yield a valid signature. However, for an encryption scheme, the workload usually is too high. Hence, this modification cannot be used to obtain such a system. In any case: in connection with the HFE-trapdoor, this modification does not prove efficient against the recent Gröbner attacks from [FJ03] as it only slightly increases the number of linearly independent monomials in P'. In addition, there is a cryptanalysis given in [DS05a] which shows that HFEv can be broken with a workload of q^v.

From a mathematical point of view, both the univariate and the multivariate variation are equivalent. This can easily be seen using the ideas of the proof of Lemma 2.4.7. Hence, it depends on the underlying trapdoor used which of the two is to be preferred in a given construction. In particular, all cryptographic attack against the univariate vinegar modification will also apply to the multivariate vinegar modification. Hence, we do not gain additional strength here.

3.2.8 Internal Perturbation: "i"

The idea of internal perturbation is due to Ding [Din04]. It was first used in connection with MIA and then denoted PMI ("Perturbated Matsumoto Imai"). One year later, the idea was extended to HFE [DS05a] and called IPHFE ("Internal Perturbation of HFE"). In both cases, an affine subspace of dimension w is used to add some kind of "noise" to the overall system. The idea is similar to HFEv (cf Section 3.2.7), but with a slight twist: while HFEv increases the number of input variables, internal perturbation does not. In a nutshell, the "old" variables $x'_1, \ldots, x'_n$ are used for two purposes: first, they span an n-dimensional vector-space in the variables $x'_1, \ldots, x'_n$ and second, they span an w-dimensional perturbation space. The advantage is that such a variation is harder to cryptanalyse. We can also see similarities to the branching modification in comparison to the STS trapdoor: for vinegar and branching, the computations were done independently while for STS and internal perturbation, the computation of the first part (first layer / perturbation polynomials) clearly influences all further computations. Hence, the attacks against branching or vinegar do not apply against STS or internal perturbation. In any case, "internal perturbation" comes in two flavours:

DEFINITION 3.2.3 *Let $\mathcal{P}', \tilde{\mathcal{P}} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ be two systems of m quadratic equations in n input variables $x'_1, \ldots, x'_n$ each. Moreover, let $s(x) : \mathbb{F}^n \to \mathbb{F}^w$ be an affine transformation,* e.g., *represented by a vector $v_s \in \mathbb{F}^w$ and a matrix $M_s \in \mathbb{F}^{n \times w}$ where the matrix M_s has rank w. We denote the output of $s(x)$ by $z' \in \mathbb{F}^w$,* i.e., *we have $z' := s(x)$ and call the components $z'_1, \ldots, z'_w$. In addition, let $\Pi \in \mathcal{MQ}(\mathbb{F}^w, \mathbb{F}^m)$ be a system of m quadratic equations in w input variables $z'_1, \ldots, z'_w$ each with components $\pi_1, \ldots, \pi_m$. Then we call*

$$\mathcal{P}' := \begin{cases} p'_1 & := & \tilde{p}_1(x'_1, \ldots, x'_n) & + & \pi_1(z'_1, \ldots, z'_w) \\ & \vdots & & & \\ p'_m & := & \tilde{p}_m(x'_1, \ldots, x'_n) & + & \pi_m(z'_1, \ldots, z'_w) \end{cases}$$

a multivariate internally perturbated *$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations.*

Definition 3.2.4 *Let $\mathbb{E}$ be an n-dimensional extension field over $\mathbb{F}$. Moreover, let $\hat{P}(X') \in \mathbb{E}[X']$ be a central equation in univariate representation (cf Lemma 2.4.5). In addition, let $s(x) : \mathbb{F}^n \to \mathbb{F}^w$ be an affine transformation represented by an $(n \times w)$-matrix of rank w and a vector of dimension w. We denote the output of $s(x)$ by $z'_1, \ldots, z'_w \in \mathbb{F}$ and we have $Z' := \phi^{-1}((z'_1, \ldots, z'_w, 0, \ldots, 0))$. In addition, let*

$$F(Z') := \sum_{0 \le i \le j < n} \hat{C}_{i,j} {Z'}^{q^i + q^j} + \sum_{i=0}^{n-1} \hat{B}_i {Z'}^{q^i} + \hat{A}$$

be a quadratic function with coefficients $\hat{C}_{i,j}, \hat{B}_i, \hat{A} \in \mathbb{E}$. Then we call

$$P'(X', Z') := \tilde{P}(X') + F(Z')$$

a univariate internally perturbated *$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations.*

As we see, in both cases the perturbation functions Π and F depend on a rather small perturbation subspace of dimension w. In addition, we do not require any trapdoor for these two functions but select their coefficients at random. Hence, we expect a workload of $O(q^w)$ for inverting the new central equation $\mathcal{P}'$. But for q^w small (*e.g.*, $q = 2$ and $w = 4 \ldots 6$), this is feasible.

At first glance, it is not obvious which of the two forms is more secure or efficient and hence advisable for the construction of public key systems. So we need the following

Lemma 3.2.5 *For every* multivariate *internally perturbated $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations, there is a* univariate *internally perturbated $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic system of equations and vice versa. Hence, both kinds of internal perturbation are equivalent from a cryptanalytic point of view.*

Proof. We use the notation from definitions 3.2.3 and 3.2.4. The overall proof is similar to the proof of Lemma 2.4.7.
$\Rightarrow$: We start with $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ and $\Pi \in \mathcal{MQ}(\mathbb{F}^w, \mathbb{F}^n)$. Our goal is to compute the corresponding univariate representation of both. This is feasible, using Theorem 2.4.9. By construction, we obtain an internal perturbation function F and a univariate polynomial P'.
$\Leftarrow$: As for the previous proof, we use Theorem 2.4.9 — but this time to obtain a multivariate representation instead of a univariate representation. The only question to answer is if our perturbation polynomials $\pi_1, \ldots, \pi_n$ depend on all n components $z'_1, \ldots, z'_n$ of Z' or only on the subset $z'_1, \ldots, z'_w$. Theorem 2.4.9 does not guarantee the latter. However, we observe that the perturbation variables $z'_1, \ldots, z'_w$ can be expressed as $R : \mathbb{F}^n \to \mathbb{F}^w$ with $R(z'_1, \ldots, z'_n) :=$

$(z'_1, \ldots, z'_w, 0, \ldots, 0)$, using the reduction from Lemma 2.4.7. Hence, the effect of the univariate perturbation function $F(Z)$ is equal to $\phi(F(\phi^{-1}(R(s(x)))))$ for all input vectors $x \in \mathbb{F}^n$. Using Theorem 2.4.9, this can be rewritten as $\Pi(R(s(x)))$ for some system of polynomials $\Pi \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$. Taking the reduction $R(\cdot)$ "into" the polynomial functions Π shows that they do not depend on the input variables $z'_{w+1}, \ldots, z'_n$, *i.e.*, we have $\Pi \in \mathcal{MQ}(\mathbb{F}^w, \mathbb{F}^m)$. Hence, the polynomial vector $\Pi = (\pi_1, \ldots, \pi_n)$ has the required form. □

By now, the "i" modification has been used with MIA (multivariate version) and HFE (univariate version). By the time of writing, we do not see any benefit when combining it with UOV or STS. However, in mixed schemes this may be different. We want to note that MIA has been broken in [FGS05], using ideas from differential cryptanalysis to "separate" the MIA scheme from the "noise" of the internal perturbation.

3.2.9 Homogenising: "h"

By taking a fresh look at the two modifications vinegar variables "v" from Section 3.2.7 and internal perturbation "i" from Section 3.2.8, we can develop a new generic modifier. We use the ideas of vinegar variables, but introduce only linear equations of degree 1 to be multiplied with the linear terms, and homogeneous equations of degree 2 to replace the constant terms. The overall result are homogeneous equations of degree 2, regardless of the trapdoor used. Hence, we have a way of saving a total of $m(1+n)$ coefficients by dropping the constant and the linear terms. As the security of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations lies in the quadratic and not the other terms, the overall security of the corresponding scheme does not degenerate with this modification. To the knowledge of the author, the "h" modification has not been proposed before. Formally, we can write this modification as follows:

DEFINITION 3.2.6 *Let $\mathbb{F}$ be a finite field $\mathbb{F}$, $h \in \mathbb{N}$ the number of homogenising variables, and $\tilde{\mathcal{P}} \in \mathcal{MQ}(\mathbb{F}^{\tilde{n}}, \mathbb{F}^m)$ a polynomial vector over $\mathbb{F}$. Moreover, let $z'_1, \ldots, z'_h$ be new variables which depend linearly on the input variables $x_1, \ldots, x_n$. The central map depends on the variables $x'_1, \ldots, x'_{\tilde{n}}$ for $\tilde{n} \le n$. Then define the central equation as*

$$\begin{aligned}
p'_1(x'_1, \ldots, x'_{\tilde{n}}) \quad &:= \quad \sum_{1 \le j \le k \le \tilde{n}} \gamma_{1,j,k} x'_j x'_k \\
&\qquad + \sum_{j=1}^{\tilde{n}} \beta'_{1,j}(z'_1, \ldots, z'_h) x'_j + \alpha'_1(z'_1, \ldots, z'_h) \\
&\vdots
\end{aligned}$$

$$
\begin{aligned}
p'_i(x'_1, \ldots, x'_{\tilde{n}}) &:= \sum_{1 \le j \le k \le \tilde{n}} \gamma_{i,j,k} x'_j x'_k \\
&\quad + \sum_{j=1}^{\tilde{n}} \beta'_{i,j}(z'_1, \ldots, z'_h) x'_j + \alpha'_i(z'_1, \ldots, z'_h) \\
&\vdots \\
p'_m(x'_1, \ldots, x'_{\tilde{n}}) &:= \sum_{1 \le j \le k \le \tilde{n}} \gamma_{m,j,k} x'_j x'_k \\
&\quad + \sum_{j=1}^{\tilde{n}} \beta'_{m,j}(z'_1, \ldots, z'_h) x'_j + \alpha'_m(z'_1, \ldots, z'_h)
\end{aligned}
$$

Here we have $A' \in \mathcal{MQ}(\mathbb{F}^v, \mathbb{F}^m)$ for $A' = (\alpha'_1, \ldots, \alpha'_m)$ with A' being homogeneous and the polynomial vector $B' \in Hom^{-1}(\mathbb{F}^v, \mathbb{F}^{m\tilde{n}})$ with coefficients $B' =: (\beta'_{1,1}, \beta'_{1,2}, \ldots, \beta'_{m,\tilde{n}})$. Then we say the central polynomial $\mathcal{P}'$ is in multivariate homogeneous *shape.*

This definition is quite similar to the definition of the vinegar modifier (cf Section 3.2.7), but with a slight twist: first, we ask for homogeneous rather than non-homogeneous equations $\alpha'_i, \beta'_{i,j}$, and second, we did not fix the source of the new variables $z_1, \ldots, z_h$ yet. Here, we may either use internal variables (cf Section 3.2.8) or "external" variables (cf Section 3.2.7). Given the cryptanalytic results previously achieved against the "v" modification of HFE, we prefer the use of internal variables. The corresponding modification will be denoted by "h". Obviously, we have $n = \tilde{n}$ and $h \le n$ here. In the case of external variables as for the "v" modification, we denote this variation "$\tilde{\text{h}}$". In this case we obtain $n = \tilde{n} + h$ as relationship between the new, the old, and the homogenising variables. We want to stress that we believe that internal variables are better suited for the purpose of homogenising the public key. In addition we want to point out that the homogenising modification only makes sense if the public key has not been constructed in a way that it is already homogeneous. In most cases, there is no need for this modification as it is possible to restrict the private key accordingly. However, in cases where we need linear terms for one reason or another, this modification proves useful to obtain a smaller public key.

3.2.10 Masking: "m"

The idea of masking variables has been developed in [Wol02a, sections 4.9 and 4.10]. It is the inverse idea to the "f" modification: instead of reducing the number of input variables, this number is increased. This is realized by changing the initial affine transformation to $\tilde{S} : \mathbb{F}^n \to \mathbb{F}^{\tilde{n}}$ for $n > \tilde{n}$ and $\tilde{S} \in \text{Aff}^{-1}(\mathbb{F}^n, \mathbb{F}^{\tilde{n}})$, *i.e.*, a surjective transformation from the vector space $\mathbb{F}^n$ to the vector space $\mathbb{F}^{\tilde{n}}$. This new transformation $\tilde{S}$ can be realized using a matrix $M \in \mathbb{F}^{\tilde{n}\times n}$ of rank $\tilde{n}$ and a vector $v \in \mathbb{F}^{\tilde{n}}$, cf Section 2.2.3. Masking increases the number of variables at the expense of the number of equations. In particular, such a system cannot be a bijection anymore. But as inverting an affine transformation is usually much faster than inverting the central system $\mathcal{P}'$, this modification can be used for the construction time efficient $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems.

The effect of this transformation is rather limited when considering, *e.g.*, the attack of [KS99]. However, in attacks which mainly depend on the number of input variables (*e.g.*, Gröbner attacks), such a modification may be worthwhile. However, as this modification has not been systematically studied its security is an open problem.

3.3 Discussion

In this chapter, we introduced constructions for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic trapdoors and showed how the schemes known so far can be grouped into a taxonomy of only four basic schemes (UOV, STS, MIA, and HFE), using 10 modifiers, cf Table 3.1. We see that there are far more modifiers than basic schemes. So to increase the number of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes, we suggest to concentrate on finding new basic trapdoors rather than new modifiers. However, given that all known trapdoors are rather old, we are not sure if many more basic trapdoors do exist.

Table 3.1: Modifiers for $\mathcal{MQ}$-schemes

Symbol	Long Name	Page	Security
-	Minus	43	secure
+	Plus	44	mostly no effect
/	Subfield	47	insecure
$\perp$	Branching	47	insecure
f	fixing	49	open
h	homogenising	55	no effect
i	internal	53	open
m	masking	56	open
s	sparse	50	open
v	vinegar	50	slightly more secure

Obviously, the taxonomy developed in this thesis can now be used to obtain new and interesting schemes. However, we urge the developers of such schemes not to combine all modifiers and trapdoors available in one scheme but to use as few as possible: if such a scheme is well designed, it will withstand cryptographic attacks while a complex scheme may distract the attention both of the cryptanalyst *and* the designer of the scheme from the real weaknesses hidden in this new construction. Moreover, each designer should make clear the rationale behind the choices made. This way it becomes much easier for the cryptographic community to evaluate the strength of the new proposals.

Apart from this, $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations have very nice properties when used in restricted environments and can be used as cryptographic primitives for signing applications. By now, the existence of secure *and* efficient encryption primitives based on the $\mathcal{MQ}$-problem is an open question. However, when we look at the authors and dates of the publications in the bibliography, we see that more and more people get interested in this subject. Hence, we may expect such a secure encryption scheme soon. We now move on to the cryptanalysis of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes. This allows us to discover, why all attempts for secure encryption schemes failed so far.

Chapter 4

Cryptanalysis of $\mathcal{MQ}$-schemes

After outlining the basic $\mathcal{MQ}$-trapdoors and their modifiers in the previous chapter, we now move on to a description of cryptographic results on these trapdoors and their combinations. Before moving on to concrete attacks, we first classify two types of attacks.

Our own achievement in this chapter are the cryptanalysis of UOV with Gröbner bases. Moreover, we introduced the affine approximation attack against $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems and performed a cryptanalysis of the STS class.

4.1 Types of Attacks

Definition 4.1.1 *Given $(x, y) \in \mathbb{F}^n \times \mathbb{F}^m$ a pair message/ciphertext or signature/message and k a public key for an $\mathcal{MQ}$-trapdoor. We want to stress that this notation is correct: in Section 2.3, we denoted y the input and x the output for signature generation. Hence (x, y) is indeed a signature/message pair. Now we denote the following problems:*

1. *Inversion problem: Recover x for given y and public key $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$.*
2. *Key recovery problem: Recover the private key $(S, \mathcal{P}', T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^m)$ for a given public key $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$.*

The attacks related to these two problems are called inversion attack and key recovery attack, respectively.

The key recovery problem is certainly more general than the inversion problem, as every attack which is able to reveal the private key $(S, \mathcal{P}', T)$ for a given public key $\mathcal{P}$ can be used to obtain the $x \in \mathbb{F}^n$ for any given $y \in \mathbb{F}^m$. So every key recovery attack is also an inversion attack.

Before dealing with the concrete attacks on specific $\mathcal{MQ}$-trapdoors, we will first show a general inversion attack which works against *all* $\mathcal{MQ}$-trapdoors.

4.2 Generic Linearisation Attack

Here we describe a very general attack against all public key systems which use $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations as their public key. To the knowledge of the author, it has first been described in [Pat96b, Sect. 3]. Here, we assume that we know $y, \hat{y} \in \mathbb{F}^m$ and some difference $\Delta \in \mathbb{F}^n$ with $\Delta =: (\delta_1, \ldots, \delta_n)$. Now we have $y = \mathcal{P}(x)$ and $\hat{y} = \mathcal{P}(x + \Delta)$ for some unknown vector $x \in \mathbb{F}^n$. We subtract the two equations $y = \mathcal{P}(x)$ and $\hat{y} = \mathcal{P}(x + \Delta)$ component-wise, and get

$$\begin{aligned}
y_i - \hat{y}_i &= p_i(x_1, \ldots, x_n) - p_i(x_1 + \delta_1, \ldots, x_n + \delta_n) \\
&= \gamma_{i,1,1}(x_1^2 - x_1^2 + 2x_1\delta_1 - \delta_1^2) + \\
&\quad +\gamma_{i,1,2}(x_1x_2 - x_1x_2 - x_1\delta_2 - x_2\delta_1 - \delta_1\delta_2) + \cdots + \\
&\quad +\gamma_{i,n,n}(x_n^2 - x_n^2 + 2x_n\delta_n - \delta_n^2) \\
&\quad +\beta_{i,1}(x_1 - x_1 - \delta_1) + \ldots + \beta_{i,n}(x_n - x_n - \delta_n) + (\alpha_i - \alpha_i) \\
&= +\gamma_{i,1,1}(2x_1\delta_1 - \delta_1^2) + \gamma_{i,1,2}(-x_1\delta_2 - x_2\delta_1 - \delta_1\delta_2) + \cdots + \\
&\quad +\gamma_{i,n,n}(2x_n\delta_n - \delta_n^2) \\
&\quad -\beta_{i,1}\delta_1 - \ldots - \beta_{i,n}\delta_n
\end{aligned}$$

for $1 \leq i \leq m$. This yields a linear system of equations in the unknowns $x_1, \ldots, x_n \in \mathbb{F}$. A solution can therefore be computed in polynomial time, *e.g.*, by Gaussian elimination. This attack falls in the class of inversion attacks.

This attack can be avoided by padding the vector x with random elements of $\mathbb{F}$ or by introducing a linearly resistant permutation (*e.g.*, AES with a publicly known key). As this attack is applicable to all $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems, it must be kept in mind when using them in practice.

4.3 Cryptanalysis of UOV

As UOV is the easiest of the schemes developed so far, we start with outlining some cryptanalytic results on this trapdoor. This section is based on [BWP05] which is joint work with An Braeken.

4.3.1 The Kipnis and Shamir Attack

We start with the key recovery attack of Kipnis and Shamir against the *Balanced* Oil and Vinegar scheme, *i.e.*, we have the number of oil variables equal to the number of vinegar variables here, cf Section 3.1.1 for the terminology of the UOV class. Here, the attacker gets access to the public key and has to compute an equivalent copy of the private key. The main idea of this attack is to separate the oil and the vinegar variables. This way, an attacker is in the same position as a legitimate user and can hence forge arbitrary signatures. The attack is very efficient for all $v \leq m$. We describe it here for $o = v = m$ and thus $2m = n$.

For the attack, we only take into account the quadratic terms of the private $\mathcal{P}'$ and the public $\mathcal{P}$ equations. We use the ideas previously outlined in Section 2.2.4: in odd characteristic, we can uniquely represent the private key equations (resp. public key equations) by $x^t P_i' x$ (resp. $x'^t P_i x'$) for $1 \leq i \leq m$, where P_i' and P_i are symmetric matrices (here ^{t} denotes transposition). For even characteristic, the unique symmetric matrices $P_i' + P_i'^t$ and $P_i + P_i^t$ where P_i' and P_i are upper-triangular matrices belonging to $\mathbb{F}^{m \times m}$ are considered. For simplicity, we denote these matrices again by P_i' and P_i, $1 \leq i \leq m$, for the private key polynomials p_i' and the public key polynomials p_i, respectively.

Note that because of the special structure of the private equations $\mathcal{P}'$, all matrices P_i' for $1 \leq i \leq m$ have the form:

$$P_i' = \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix},$$

where $0, A_i, B_i, C_i$ are submatrices of dimension $m \times m$. Because $\mathcal{P} = \mathcal{P}' \circ S$, we obtain

$$P_i = M_S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} M_S^T.$$

It is clear that each P_i' maps the subspace $x_{m+1}, \cdots, x_{2m}$ (oil subspace) to the subspace $x_1 = \cdots = x_m = 0$ (vinegar subspace). If P_j' is invertible, we can then conclude that each $P_i' P_j'^{-1}$ maps the oil subspace to itself. Consequently the image of the oil subspace under S, called the subspace O, is a common eigenspace for each $P_i P_j^{-1}$ with $1 \leq i < j \leq m$. In [KS98, Sect. 4], Shamir and Kipnis describe two very efficient algorithms for computing the common eigenspace O of a set of transformations. Picking a subspace V for which $O + V = \mathbb{F}^m$ allows us to separate the oil and the vinegar variables. This way, we obtain an equivalent copy of the private key $(\mathcal{P}, S)$ and hence, the Kipnis-Shamir attack is a key recovery attack. The overall attack complexity is $O(m^4)$ for $m \in \mathbb{N}$ the number of equations. As we have $n = 2m$ in this setting, the overall attack complexity can be described without reference to the number of variables n.

In [KPG99, Sect. 4], an extension based on a probabilistic approach of the previous attack is described which also works for $v > m$ (or $n > 2m$) with complexity $O(q^{v-m-1}m^4) = O(q^{n-2m-1}m^4)$.

Application against the Parameters from [KPG03, Sect. 14]. In order to avoid the birthday paradox, [KPG99, Sect. 8] describes a modification of UOV which fixes the linear terms of the public equations depending on the message M. This way, it is no longer possible to obtain a collision for different messages $M_1 \neq M_2$ and the same public key, as this public key now also depends on the message M. We consider this construction to be secure and therefore refer to [KPG99, Sect. 8] for a detailed description. However, its application in [KPG03, Sect. 14], Example 4 is flawed. In order to derive a smaller public key, the authors use the subfield modification "/" (cf [KPG03, Sect. 10]). As outlined in Section 3.2.3, all coefficients in the affine transformation S and the system of private polynomials $\mathcal{P}'$ are not chosen from the field $\mathbb{F}$ but from a strictly smaller subfield $\tilde{\mathbb{F}}$. This way, the public key $\mathcal{P}$ will only have coefficients from $\tilde{\mathbb{F}}$ as $\mathcal{P} = \mathcal{P}' \circ S$ and subfields are closed under addition and multiplication. Thus, we derive a public key which is a factor of $(\log|\tilde{\mathbb{F}}|/\log|\mathbb{F}|)$ smaller than the original key.

[KPG03, Example 4] suggests $\mathbb{F} = \mathrm{GF}(16)$, $\tilde{\mathbb{F}} = \mathrm{GF}(2)$, $m = 16$, $v = 32/48$ and obtain a public key with 2.2kB/4 kB — this is 4 times smaller than without this trick. However, we can apply the attack from the [KPG99, Sect. 4] (see Section 4.3.1, above) against the UOV system over $\tilde{\mathbb{F}} = \mathrm{GF}(2)$. This is possible as the Kipnis-Shamir attack does not take *linear* terms into account but only quadratic terms. The crucial point is that the linear terms are from GF(16) while the quadratic terms are from a subfield isomorphic to GF(2). As soon as we derived an isomorphic copy of the private key $(\mathcal{P}, S)$ over GF(2), we can translate it to GF(16) and are now in the same position as a legitimate user. In particular, we can do all computations necessary to translate the linear parts of the public key (over GF(16)) to the corresponding private key (now, also over GF(16)). As we have $q = 2$, the attack complexity is $2^{32-16-1} \cdot 16^4 = 2^{31}$ or $2^{48-16-1} \cdot 16^4 = 2^{47}$ and therefore far less than the claimed security level of 2^{64}. The above application of the Kipnis-Shamir attack has been found by the author and was published in [BWP05].

Remark: Although the algorithms from [CGMT02] achieve a lower running time, they are not applicable in this case: they are only able to solve a given instance of an $\mathcal{MQ}$-problem, *i.e.*, they do not allow key recovery but only inversion. For this attack, we need the fact that we actually derive a valid private key of the UOV-system, *i.e.*, a key recovery algorithm.

4.3.2 Attacks using Gröbner Basis Algorithms

A second inversion attack on UOV, first described in [BWP05], is inspired by the article of Daum, Felke, and Courtois [CDF03] which outlines a way of attacking HFE with Gröbner Basis algorithms. The attack of [CDF03] proves particularly efficient against the HFE system with the vinegar "v" modification and partly against the minus "-" modification, too.

In [BWP05] this inversion attack is also applied against the UOV class. In a nutshell, the authors conclude that the attack works well if the number of variables is roughly equal to the number of equations. Unfortunately, this is not the case for secure choices of parameters, *i.e.*, with $n = 3m \dots 4m$. In particular, attacking UOV with Gröbner base algorithms is less efficient than exhaustive search, even for a moderate number of variables like $n = 32$. In particular, the authors conclude from their experiments that the workload of Gröbner base attacks exceeds 2^{64} for 38 or more equations in the finite field 2. Even for this choice of parameters, we expect a brute force attack to be successful after around 2^{38} tries.

4.3.3 Exploiting the Existence of Affine Subspaces

This attack extends the attack of Youssef and Gong [YG01] against the Imai and Matsumoto Scheme B [IM85]. It exploits the fact that a cryptosystem can be approximated by several affine equations. The original attack was designed for fields of even characteristic. The attack described in this section is generalised to all characteristics.

In a nutshell, the attack assembles several points belonging to the same affine subspace W. Having w points $x_1, \dots, x_w \in \mathbb{F}^n$ for which UOV is affine, a function $F(x) = Ax + b$ can be used to describe the output of UOV. In order to launch the attack, we first compute the corresponding $y_i = UOV(x_i)$ for $1 \le i \le w$ and $y_i \in \mathbb{F}^m$. With this knowledge, we can determine for any given y' if it belongs to the subspace W and — if this is the case — compute a vector $a \in \mathbb{F}^w$ with $y' = \sum_{i=1}^{w} a_i y_i$. As the subspace W is affine, we can then determine the corresponding $x' \in \mathbb{F}^n$ as $\sum_{i=1}^{w} a_i x_i$. In the following section, we will present several ways of computing the points x_i, *i.e.*, to determine one or several subspaces W.

For UOV, there exist approx. q^v subspaces of dimension $o = m$ on which UOV is affine. Moreover, all these subspaces are disjunct. If we can find $(o+1)$ linearly independent points of the same subspace, we completely break the scheme for this subspace. If we find fewer, *e.g.*, w points, we have at least covered q^w points of the corresponding subspace W. Repeating the search for $(o+1)$ points q^v times, we break the whole scheme. Note that it is sufficient for the signature forgery of a given $y \in \mathbb{F}^m$ if we know **one** subspace W for which $y \in W$. Therefore, we do not need to know all q^v subspaces but only a small number for forging any signature $x \in \mathbb{F}^n$ for a given message $y \in \mathbb{F}^m$ with high probability. We recall that $(x, y) \in \mathbb{F}^n \times \mathbb{F}^m$ was defined as a signature/message pair. Hence the task is to compute $x \in \mathbb{F}^n$ for given $y \in \mathbb{F}^m$.

In order to search for points which are in the same subspace, we use the following observation: if the 3 points $R_1, R_2, R_3 \in \mathbb{F}^n$ are in the same affine subspace with respect to UOV, the following condition has to be satisfied:

$$UOV(R_1) - UOV(R_2) - UOV(R_3) + UOV(-R_1 + R_2 + R_3) = 0\,. \quad (4.1)$$

We check the validity of the above equation by computing

$$AR_1{+}v{-}AR_2{-}v{-}AR_3{-}v{+}A(-R_1{+}R_2{+}R_3){+}v = A(R_1{-}R_1{+}R_2{-}R_2{+}R_3{-}R_3)$$

Using this property, we can determine points of the same affine subspace by repeating the heuristic algorithm described in Figure 4.1 several times. The corresponding algorithm for even characteristic has been described in [YG01].

Input: point R_1, public key $\mathcal{P}$ of UOV
Output: A pair (R_1, R_2) of points which belong to the same affine subspace
repeat
  $pass \leftarrow 0$
  $trials \leftarrow 0$
  $R_2 \leftarrow \text{Random}(\mathbb{F}^n)$
  $\delta_x \leftarrow -R_1 + R_2$
  repeat
    $trials \leftarrow trials + 1$
    $R_3 \leftarrow \text{Random}(\mathbb{F}^n)$
    $R_4 \leftarrow \delta_x + R_3$
    $\delta_y \leftarrow UOV(R_1) - UOV(R_2) - UOV(R_3) + UOV(R_4)$
    if $(\delta_y = 0)$ **then** $pass \leftarrow pass + 1$
  until $(pass > threshold)$ or $(trials > q^v \cdot threshold)$
until $(pass > threshold)$ or $(trials > q^v \cdot threshold)$
OUTPUT (R_1, R_2)

Figure 4.1: Algorithm to find a pair of points in the same affine subspace for which UOV is affine

By repeating this algorithm often enough for a fixed point R_1, we obtain $(o+1)$ linearly independent points of one affine subspace. The complexity of the algorithm will be roughly $O(q^{2v})$, according to the probability that R_1, R_2 and R_3 belong to the same affine subspace.

This attack can be improved using the relation

$$UOV(R_1) + UOV(R_2) - UOV(R_1 + R_2) = b \quad (4.2)$$

for some fixed $b \in \mathbb{F}^m$. As soon as we find a triple $(R_1, R_2, R_3) \in (\mathbb{F}^n)^3$ of points which yield $\delta_y = 0$ in Algorithm 4.1, we use (4.2) to check if all of them yield the same constant b. If this is the case, we can conclude with probability q^{-2m} that all three points belong to the same subspace. At this point, we can change to another algorithm: instead of checking triples, we now check pairs. If the pair (R_1, R') yields the constant b, we found a new candidate belonging to the same subspace as R_1. Using the other points found so far, we can increase the probability that R' is genuine further by q^{-m} with each point we try. We summarise this algorithm:

1. Find a triple $(R_1, R_2, R_3) \in (\mathbb{F}^n)^3$ which satisfies (4.1).
2. Using this triple and (4.2), determine the value of the constant $b \in \mathbb{F}^m$.
3. Use (4.2) to find more points $R' \in \mathbb{F}^n$ in the same subspace.
4. As soon as $(o + 1)$ points $R \in \mathbb{F}^n$ are known, determine matrix A by Gaussian elimination.

The running time of this algorithm is $O(q^{2v} + (n - v)q^v)$ on average as we chose the points R_2 and R_3 independently from the point R_1 in the first step and R' also independently from R_1. The overall running time to find a total of $(o + 1)$ points in the same subspace becomes therefore $O(q^{2v})$ as $O(oq^v)$ is negligible in comparison to $O(q^{2v})$.

We are able to speed up Algorithm 4.1 from Section 4.3.3 if we can spend some memory and also have $m > v$, *i.e.*, we do have "enough" equations in relation to the dimension v of the affine subspaces to be found. This is certainly not true for UOV — here we have typically $m < v$ or even $m < 2v$ (see above). However, for other multivariate quadratic systems, this condition may hold. In particular, it is the case for System B of Matsumoto-Imai, cf [YG01]. We therefore present two ways of speeding up Algorithm 4.1. We explain it for the example of UOV to simplify the discussion but want to stress that it also works against System B or any other multivariate quadratic system which allows affine approximations of small dimension.

Triple Algorithm

If we can spend $O(kq^{2v})$ of memory for some small k (e.g., $10 \le k \le 20$) and also have $m > v$, we can achieve a time/memory-tradeoff for finding **all** subspaces in UOV by using the following technique. In the precomputation phase, we evaluate random pairs $(R_1, R_2) \in_R \mathbb{F}^n \times \mathbb{F}^n$ using (4.2). The probability for each of these pairs to have points in the same affine subspace is q^{-v} (birthday paradox). Moreover, we know that two points in the same subspace will yield the same constant $b \in \mathbb{F}^m$. On the other hand, two points which are not in the same subspace will yield a random value $v \in \mathbb{F}^m$. The probability for each of these values to occur is q^{-m} with $m > v$. As we were dealing with a total of kq^{2v} pairs, we do not expect two random values $v_1, v_2 \in \mathbb{F}^n$ to occur more often than, say, $\frac{k}{2}$ times. Therefore, all values occurring more often than $\frac{k}{2}$ are constants b with very high probability. Checking the points in the corresponding pairs using (4.1), we can even distinguish pairs of different subspaces which yield the same constant b. After this precomputation step, we can check for each point $R' \in \mathbb{F}^n$ to which of the q^v subspaces it belongs, using $O(q^v)$ computations on average. After $O(oq^v)$ trials, we have $(o+1)$ points for each subspace and can therefore determine the matrix $A \in \mathbb{F}^{n\times n}$ and the vector b for the affine equation $F(x) = Ax + b$. The above algorithm can be summarised as follows:

1. Use Equation 4.2 on kq^{2v} random pairs $(R_1, R_2) \in_R \mathbb{F}^n \times \mathbb{F}^n$ and store triples $(b, R_1, R_2) \in \mathbb{F}^m \times \mathbb{F}^n \times \mathbb{F}^n$
2. Check for each value $b_i \in \mathbb{F}^m$ how often it occurs in the stored list
3. For values b_i which occur at least $\frac{k}{2}$ times, use (4.1) to check whether the corresponding triples belong to the same affine subspace.
4. Use (4.2) to determine more points $R' \in \mathbb{F}^n$ for each of these subspaces.

The overall running time of this algorithm is $O(q^{2v})$. However, the drawback is that we need an amount of memory that grows exponentially with $2v$. Therefore, it seems to be advisable to use the following algorithm $O(q^v)$ times instead. This leads to the same overall running time but requires less memory, namely only $O(q^v)$.

Pair Algorithm

Using a similar idea, we can also reduce the running time for finding the corresponding subspace W for **one** given point $R_1 \in \mathbb{F}^n$. However, we need $O(kq^v)$ memory for some small k, e.g., $10 \le k \le 20$. In this setting, we evaluate pairs (R_1, R_2) for randomly chosen $R_2 \in_R \mathbb{F}^n$ and store the corresponding triples $(b, R_1, R_2) \in \mathbb{F}^m \times (\mathbb{F}^n)^2$. With a similar argument as for the previous algorithm, we expect a random distribution for the values $b_i \in \mathbb{F}^m$ — except if the pair (R_1, R_2) for given R_1, R_2 is in the same vector space W. This event occurs with probability q^{-v}. Therefore, we can assume that the correct value b will occur k times on average and with very high probability at least $\frac{k}{2}$ times. As soon as we have found this value b, we can look for more values R' which satisfy (4.2). The overall running time of this algorithm is $O(kq^v)$ for the first step and $O(oq^v)$ for the second step, *i.e.*, $O(q^v)$ in total. However, the drawback is that we need an amount of memory that grows exponentially with v.

Both speed-ups do no longer work for $v, m = \frac{n}{2}$ as the "gap" between q^{-v} and q^{-m} no longer exists. Therefore, we cannot distinguish anymore between values b and random values.

The advantage of the affine approximation attack against UOV is that we know exactly the structure of these affine subspaces. In addition, all these affine subspaces are disjunct. This was not the case for System B from Matsumoto-Imai [IM85]. Theoretical predictions of the running time of the algorithm were therefore more difficult.

4.3.4 Discussion

We saw in the previous section that UOV with $o = v$, *i.e.*, as many oil as vinegar variables, can be attacked efficiently using an idea of Kipnis and Shamir. In fact, their idea has been extended to the case of $v > o$ and has been used by the author to successfully attack a special choice of parameters for UOV, cf Section 4.3.1. In addition, the vulnerability of the UOV class against Gröbner attacks has been investigated. It turns out that UOV is quite resistant against this type of attack — even when using the Daum-Felke-Courtois strategy. A new class of attacks is the affine approximation attack. While it works well for v small, this is unfortunately not the case for UOV.

A different type of attack has been discussed in [CGMT02]. Here, special purpose algorithms for $n \gg m$ have been proposed. While they do not use the special structure of the UOV trapdoor, they make use of the fact that secure UOV requires $v \geq 2o$, and hence, we have $n \geq 3m$.

4.4 Cryptanalysis of STS

In the previous section, we described several attacks on the UOV class. We now move on to the STS class, which has been developed by An Braeken and the author. In addition, the following cryptanalysis is also the result of a collaboration with An Braeken.

All attacks described in this section deal with regular STS (rSTS), *i.e.*, we have the same number of new variables and new equations in each step. In symbols: $n_1 = \ldots = n_L = m_1 = \ldots = m_L = r$ for some step-width $r \in \mathbb{R}$ and $m = n = Lr$, cf Section 3.1.2 for this notation. We want to stress that the cryptanalysis also works for general STS (gSTS), but the details become more cumbersome. For the sake of clarity, we decided to concentrate on rSTS in this section. In this context, we developed both an inversion attack (cf Section 4.4.3) against rSTS, which recovers for given ciphertext $y \in \mathbb{F}^m$ the corresponding message $x \in \mathbb{F}^n$. In the key recovery attack (cf Section 4.4.4), we build an equivalent version of the private key, denoted

$(\tilde{S}, \tilde{\mathcal{P}}', \tilde{T}) \in \mathrm{Hom}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathrm{Hom}^{-1}(\mathbb{F}^m)$. Using $(\tilde{S}, \tilde{\mathcal{P}}', \tilde{T})$, the attacker is now in the same position as the legitimate user for deciphering a given ciphertext y or forging a signature on it. For both attacks, we first need some observations on kernels of matrices.

4.4.1 Chain of Kernels

Before moving on to these kernels, we want to recall that one can express any $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomial as an $(n \times n)$-matrix over $\mathbb{F}$, cf Section 2.2.4. This is both true for a public key polynomial p_i, but also a private key polynomial p'_i. Following the notation outlined in the previous sections, we denote the corre-

$$\begin{pmatrix}
\gamma'_{i,1,1} & \gamma'_{i,1,2}/2 & \cdots & \cdots & \gamma'_{i,1,rl}/2 & 0 & \cdots & 0 \\
\gamma'_{i,2,1}/2 & \gamma'_{i,2,2} & & & \gamma'_{i,2,rl}/2 & 0 & & 0 \\
\vdots & \vdots & \ddots & & \vdots & \vdots & & \vdots \\
\gamma'_{i,rl-1,1}/2 & \gamma'_{i,rl-1,2}/2 & & \gamma'_{rl-1,rl-1} & \gamma'_{i,rl-1,rl}/2 & 0 & \cdots & 0 \\
\gamma'_{i,rl,1}/2 & \gamma'_{i,rl,2}/2 & \cdots & \gamma'_{rl,rl-1}/2 & \gamma'_{i,rl,rl} & 0 & & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{pmatrix}$$

Figure 4.2: Matrix Representation P'_i of the Private Key p'_i for Layer l

sponding matrices $P_i \in \mathbb{F}^{n \times n}$ and $P'_i \in \mathbb{F}^{n \times n}$, respectively. Obviously, the rank of each such matrix depends on its layer l: in each such layer, we only have input variables $x'_1, \ldots, x'_{rl}$. We may also say that all coefficients associated to variables $x'_{rl+1}, \ldots, x'_n$ are equal to 0. This idea has been summarised in Figure 4.2. The matrices P'_i have a rank of rl in each layer l for $1 \le l \le L$ and we have

$$\ker'_l = \{a' \in \mathbb{F}^n |\ a'_1 = \ldots = a'_{rl} = 0\}$$

as common kernels of the matrices P'_i for $(l-1)r < i \le lr$, see Figure 4.2 for the structure of the matrix for a given layer $l \in \mathbb{N}$. As these kernels are hidden by the linear transformation S, we also mark them with a prime $'$. Moreover, we denote by $a'_i \in \mathbb{F}$ for $1 \le i \le n$ the coefficients of the vectors $a' \in \mathbb{F}^n$.

We now study the effect of the linear transformation S, *i.e.*, the change of variables. As we have $\hat{p}_i := p'_i \circ S$ and $x' = S(x)$, we obtain $\hat{P}_i := SP'_iS^t$ in terms of the corresponding matrices. As the transformation S is invertible, we have $\mathrm{Rank}(\hat{P}_i) = \mathrm{Rank}(P'_i)$ and

$$\ker_l \quad = \quad \{a'S^{-1} \mid a' \in \mathbb{F}^n \wedge a'_1 = \ldots = a'_{rl} = 0\} \tag{4.3}$$

for the kernels of $\hat{P}_i$ for $(l-1)r < i \leq lr$ and an unknown matrix $S \in \mathbb{F}^{n \times n}$. Moreover,

$$\ker'_L \subset \ldots \subset \ker'_1 \text{ and consequently } \ker_L \subset \ldots \subset \ker_1 \ .$$

With the notation $T = (\tau_{i,j})_{1 \leq i,j \leq m}$, each individual public key matrix P_i can be expressed by

$$P_i = \sum_{j=1}^{m} \tau_{i,j}[SP'_i S^t] = \sum_{j=1}^{m} \tau_{i,j} \hat{P}_i \, .$$

The problem of finding the transformation T^{-1} and thus T has therefore been reduced to finding a linear combination of the public key (in matrix notation) which has a specific rank, *i.e.*, to a MinRank problem (cf Section 2.6.2). While the general MinRank problem is difficult on average and even $\mathcal{NP}$-complete, the above problem is defined for matrices with a very special structure. As we will see below, this special structure is actually a weakness and allows for attacks against STS schemes.

4.4.2 Recovering the Transformation T

As we saw in the previous section, it is crucial for an attack of STS schemes to recover the transformation T. In this section, we describe two algorithms which can be used for this purpose.

Attacking the High-Rank Side.

We start with an attack on the high-rank side (cf the algorithm in Figure 4.3). The overall idea of this algorithm is to exploit the step-structure of STS. To do so, we observe that a correct random guess of a row-vector in T^{-1} will lead to a condition on the rank of the linear combination of the corresponding public key equations — expressed in matrix notation. More formally and also to verify the correctness of this algorithm, we consider the following vector spaces.

Definition 4.4.1 *Define a descending chain of subspaces J_l of dimension $m - lr$ each for $1 \leq l \leq L$ as*

$$J_l \ := \ \{b'T^{-1} \mid b' \in \mathbb{F}^m \wedge b'_{lr+1} = \ldots = b'_m = 0\} \text{ for } 1 \leq l \leq L \ . \quad (4.4)$$

When picking a random element $v \in_R J_{l+1}$, we have a probability of q^{-r} that the expression $v \in J_l$ holds because of the definition of the subspaces J_l, J_{l+1}. In addition, we have two efficient methods (`matrixCheck` or `polynomialCheck`, respectively) to check whether $v \in J_l$ or $v \notin J_l$. First, we concentrate on `matrixCheck`.

Lemma 4.4.2 *The method* `matrixCheck` *will check if $v \in J_l$ and is defined by*

$$\texttt{matrixCheck}(P_1, \dots, P_m,\ v, l) \text{ returns } \textbf{\textit{true}} \text{ iff } Rank(\sum_{i=1}^{m} v_i P_i) \le lr \ .$$

Proof. For the sake of the argument, we look at the problem in the T^{-1}-space, *i.e.*, after the linear transformation T^{-1} has been applied. Using the notation from (4.4), we consider vectors b' instead of v. Hence we have

$$M := \sum_{i=1}^{m} b'_i \hat{P}_i = \sum_{i=1}^{rl} b'_i \left(S P'_i S^t \right) = S \left(\sum_{i=1}^{rl} b'_i P'_i \right) S^t \ .$$

Observing the step-wise structure of the private key polynomials p'_i we conclude that $\text{Rank}(M) \le lr$. This yields the result. □

The expected running time of the algorithm from Figure 4.3 is therefore bounded by $O(mn^3 L q^r)$: by picking at most cmq^r vectors for each layer (c being a small constant, *e.g.*, 10), we can compute the vector spaces $J_1, \dots, J_L$ with very high probability. Checking the matrix condition costs an additional factor of n^3 as we are processing matrices from $\mathbb{F}^{n \times n}$. In comparison, the running time of the other steps of the algorithm are negligible.

In characteristic 2 we may apply Dickson's theorem instead to check directly for a given polynomial if it may be reduced to a form with less variables (procedure `polynomialCheck`). Unfortunately, the proof is a bit lengthy; we therefore refer to [MS91, Sect. 15.2, Thm. 4] for both the theorem and its proof. An algorithmic version of it can be found in [CGMT02, Sec. 3.2]. The time complexity of this algorithm is there estimated to be $O(n^3)$. Therefore, the overall complexity of the above algorithm remains the same: $O(mn^3 L q^r)$.

Remark 4.4.3 *In both cases, we will not be able to recover the original transformation T but the inverse of a linear equivalent copy of it, denoted $\hat{T}$ for the inverse and $\tilde{T}$ for the linear equivalent of T. In fact, we will recover versions of T in which the rows of $\tilde{T}$ are linear combinations of the rows of T within the same layer. Using arguments from Chapter 5, we see that we actually recover one representative of an equivalence class of possible private keys, all leading to the same public key. This stresses the importance of finding equivalent keys for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes.*

Attacking the Low-Rank Side.

In the previous algorithm we constructed the linear equivalent copy of T^{-1} step-wise by means of the r basis vectors from the subspace $\tilde{J} = J_{l+1} \cap J_l$ for $l = L \dots 1$.

procedure highRankAttack($\mathcal{P}$)
  Input: $\mathcal{P}$: system of public equations
  Output: $\tilde{T}$: an equivalent copy of the transformation T
  $P_i \leftarrow$ computeMatrix(p_i); $J_L \leftarrow \mathbb{F}^m$
  for $l \leftarrow L-1$ **downto** 1 **do**
    $J_l \leftarrow \emptyset$
    repeat
      $v \in_R J_{l+1}$
      if matrixCheck($P_1, \ldots, P_m,\ v, l$) $\vee$ polynomialCheck($p_1, \ldots, p_m,\ v, l$) **then**
        $J_l \cup \leftarrow \{v\}$
    until Dimension(J_l) $\stackrel{?}{=} lr$
    $\tilde{J} \leftarrow J_{l+1} \cap J_l$
    for $i \leftarrow 1$ **to** r **do**
      RowVector($\hat{T}, lr+i$) $\leftarrow$ BasisVector($\tilde{J}, i$)
  endfor
  return $\tilde{T} \leftarrow \hat{T}^{-1}$
endproc

Figure 4.3: High-Rank algorithm for computing the transformation $\tilde{T}$ for a given public key $\mathcal{P}$

Therefore we call it an attack from the high-rank side. We now show how we can also perform an attack from the low-rank side.

For the following lemma, we use the same notation as in Definition 4.4.1 but set $J_0 := \{0 \in \mathbb{F}^m\}$, *i.e.*, the all-zero vector in an m-dimensional vector space over the ground field $\mathbb{F}$.

Lemma 4.4.4 *The subspace $\tilde{J} := J_l \cap \overline{J_{l-1}}$ where $\overline{J_{l-1}}$ denotes the complement of the vector space J_{l-1}, has dimension r and will determine r new linearly independent rows of the matrix T^{-1}.*

Proof. The proof is based on two different observations. The first one is that the kernels $\ker_i$ form a descending chain. Therefore, setting $\ker_0 := \mathbb{F}^n$, the statement $w \in \ker_l$ is true with probability q^{-r} for all $w \in_R \ker_{l-1}$ and $1 \leq l \leq L$. Second, the linear equation $\sum_{i=1}^{m} v_i(wP_i) = 0$ has q^{lr} solutions for unknown $v \in \mathbb{F}^m$ if and only if the vector w is in the kernel $\ker_l$. □

Algorithm 4.4 will therefore terminate with a correct solution $\tilde{T}$ after a total of $O(Ln^3q^r)$ steps on average. Thus it outperforms the algorithm from the previous section by a factor of m. As for the previous algorithm, we will not recover the

procedure lowRankAttack($\mathcal{P}$)
  Input: $\mathcal{P}$: system of public equations
  Output: $\tilde{T}$: an equivalent copy of the transformation T
  $P_i \leftarrow$ computeMatrix(p_i); $K_0 \leftarrow \mathbb{F}^n$; $J_0 \leftarrow \{0\}$
  for $l \leftarrow L$ **downto** 1 **do**
    repeat
      $w \in_R K_{l-1}$
      $J_l \leftarrow$ SolutionSpace($\sum_{i=1}^{m} v_i(wP_i) = 0$) for an unknown $v \in \mathbb{F}^m$
    until Dimension(J_l) $\stackrel{?}{=} lr$.
    $\tilde{J} \leftarrow J_l \cap \overline{J_{l-1}}$
    for $i \leftarrow 1$ **to** r **do**
      $\hat{t} \leftarrow$ BasisVector($\tilde{J}, i$); RowVector($\hat{T}, lr + i$) $\leftarrow \hat{t}$; $\hat{P}_{(l-1)r+i} \leftarrow \sum_{j=1}^{m} \hat{t}_j P_j$
    $K_l \leftarrow$ Kernel(P_{lr})
  endfor
  return $\tilde{T} \leftarrow \hat{T}^{-1}$
endproc

Figure 4.4: Low-Rank algorithm for computing the Transformation $\tilde{T}$ for a given public key $\mathcal{P}$

original transformation T but an equally useful variant of it.

Remark: Specialised versions of the algorithms from figures 4.3 and 4.4 can be found in [GC00] for the case of schemes with step-width 1 of the intermediate layers.

4.4.3 Inversion Attack

In the previous section, we have discussed two different approaches to recover a linear transformation $\tilde{T}$ for given public key equations. In this section, we will use $\tilde{T}$ and the polynomials $\hat{p}_i := \tilde{T}^{-1} \circ p_i$ to solve the problem $y = \mathcal{P}(x)$ for a given vector $y \in \mathbb{F}^m$, *i.e.*, for the $\mathcal{MQ}$-problem. We do so by computing a successive affine approximation of x, cf Figure 4.5. Define $K_i := \ker_i$ for $1 \leq i \leq L$.

Lemma 4.4.5 *The solutions of the inversion problem form a chain of affine subspaces* $x + <K_l>$ *— where* K_l *has dimension* $(n - rl)$ *in step* l.

PROOF. Recall that the kernels $K_i := \ker_i$ for $1 \leq i \leq L$ have the form $\ker_l = \{a'S^{-1} \mid a' \in \mathbb{F}^n \wedge a'_1 = \ldots = a'_{rl} = 0\}$. Setting $K_0 := \mathbb{F}^n$ we have

$$\tilde{K}_l = K_{l-1} \cap \overline{K_l} = \{a'S^{-1} \mid a' \in \mathbb{F}^n \wedge a'_1 = \ldots = a'_{(l-1)r} = a'_{lr+1} = \ldots = a'_n = 0\}$$

for $1 \le l \le L$. Using this observation, we can manipulate groups of r (hidden) variables x' and therefore influence the output of the polynomials $\hat{p}_i$ layer by layer. This is possible although we do not know the actual value of the secret matrix S. The statement in the lemma then follows from the fact that the polynomial system $\hat{\mathcal{P}}$ inherits the layer structure of the original private polynomial system $\mathcal{P}'$, *i.e.*, we have a descending chain of subspaces. □

Using the algorithm from above, we learn $r \log_2 q$ bits about the vector x for each level of recursion. With this inversion attack, we are now in a similar position

procedure inversionAttack($\mathcal{P}$, $\tilde{T}$, $K_1, \ldots, K_L$, y)
  Input: $\mathcal{P}$: system of public equations, $\tilde{T}$: linear transformation,
    $K_1, \ldots, K_L$: descending chain of kernels, y: target-value
  Output: X: a set of solutions for the problem $y = \mathcal{P}(x)$

  procedure recursivePart(x, l)
    if $l > L$ **then return** $\{x\}$
    $\tilde{K} \leftarrow K_{l-1} \cap \overline{K_l}$; $X \leftarrow \emptyset$
    for $\forall w \in \tilde{K}$ **do**
      if $(\hat{p}_i(x + w) \stackrel{?}{=} \tilde{y}_i \ : \ (l-1)r < i \le lr)$ **then** $X \cup\leftarrow$ recursivePart($x + w$, l)
    return X
  endproc

  $\hat{p}_i \leftarrow p_i \circ \tilde{T}^{-1} : 1 \le i \le m$
  $\tilde{y} \leftarrow y\tilde{T}^{-1}$; $K_0 \leftarrow \mathbb{F}^n$
  return recursivePart(0,1)
endproc

Figure 4.5: Inversion attack for $y = \mathcal{P}(x)$ and given $\tilde{T}$

as the legitimate user: at each level, we have to try cq^r possible vectors and to evaluate r polynomials $\hat{p}_i$ — each step costing $O(rn^2)$. In case the STS is not a bijection, we may need to branch — but this is the same situation as for the legitimate user. The only additional overhead is the computation of the complement of vector spaces and to intersect them. Both can be done in $O(n^2)$. Assuming that $\mathcal{P}$ is a bijection, one application of this inversion attack has time-complexity $O(n^2 Lrq^r)$.

4.4.4 Key Recovery Attack

The starting point of the key recovery attack (cf Figure 4.6) is the same as for the inversion attack, namely $ker_1 \supset \ldots \supset ker_L$, due to $ker'_1 \supset \ldots \supset ker'_L$. As we have computed the transformation $\tilde{T}$ in the previous step, we are able to compute the system of equations $\hat{\mathcal{P}}$, the corresponding matrices $\hat{P}_l$ and therefore their kernels for each layer $l : 1 \leq l \leq L$. Due to its internal structure, the vector space $\tilde{K} := K_{l-1} \cap \overline{K_l}$ consists of exactly r row-vectors of $\tilde{S}^{-1}$. We recover them in the for loop. As soon as we have recovered $\tilde{S}$, we apply it to the intermediate system of equations $\hat{P}$, yielding $\tilde{\mathcal{P}}$', an equivalent copy of the private key polynomials. As we will see in Chapter 5, this is actually the best we can do as each private key $(S, \mathcal{P}', T) \in \text{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \text{Aff}^{-1}(\mathbb{F}^m)$ is only a representative of a class of private keys which all lead to the same public key $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$.

In terms of complexity, the second step of the key recovery attack is dominant: we need to evaluate m polynomials with $O(n^2)$ quadratic terms each. As each quadratic term has two variables, this costs $O(n^2)$ for each term. The overall time complexity is therefore $O(mn^4)$. So depending on the value q^r, either the key recovery or the inversion attack has a lower asymptotic running time as the constants are in the same range. From our experiments, we expect constants between 10 and 100.

procedure keyRecoveryAttack($\hat{\mathcal{P}}$, $K_1, \ldots, K_L$)
  Input: $\hat{\mathcal{P}}$: system of equations; $K_1, \ldots, K_L$: descending chain of kernels
  Output: $\tilde{S}$: an equivalent copy of the secret transformation S
    $\tilde{\mathcal{P}}'$: an equivalent copy of the private key polynomials
  $K_0 \leftarrow \mathbb{F}^n$
  for l ←1 **to** L **do**
    $\tilde{K} \leftarrow K_{l-1} \cap \overline{K_l}$
    RowVector($\hat{S}$, $(l-1)r + i$) ←BasisVector($\tilde{K}, i$) $: 1 \leq i \leq r$
  $\tilde{S} \leftarrow \hat{S}^{-1}$
  $\tilde{p}'_i \leftarrow \hat{p}_i \circ \tilde{S}^{-1} : 1 \leq i \leq m$
  return $\tilde{S}, \tilde{P}'$
endproc

Figure 4.6: Structural attack for a given sequence of kernels $\text{ker}_1, \ldots, \text{ker}_L$

4.4.5 Special Instances of STS

In this section, we show that the two schemes RSE(2)PKC and RSSE(2)PKC [KS04c, KS04b], proposed by Kasahara and Sakai, are special instances of STS — and will therefore fall for the attacks discussed in the previous section. In particular, we were able to break the challenge proposed in [KS04c, Sect. 6] using an inversion attack (cf Section 4.4.3) in both cases.

RSSE(2)PKC. In RSSE(2)PKC, the private polynomials p'_i for $1 \leq i \leq r$ have a special form, namely

$$p'_{(l-1)r+i}(x') := \phi_{l,i}(x'_{(l-1)r+1}, \ldots, x'_{lr}) + \psi_{l,i}(x'_1, \ldots, x'_{(l-1)r}) \text{ for } 1 \leq l \leq L\,,$$

where $\phi_{l,i}$ and $\psi_{l,i}$ are random quadratic polynomials over $\mathbb{F}$ in r and $(l-1)r$ variables, respectively. In both cases, the constant part is omitted. To simplify the structure, the linear terms βx_i are considered to be quadratic terms βx_i^2, for all $i \in \{1, \ldots, n\}$. This may be done as RSSE(2)PKC is defined over GF(2) and we hence have $x^2 = x$ for all $x \in \mathrm{GF}(2)$.

We observe that this special construction of the private key polynomials does not affect our attacks. In particular, the maximum rank for the corresponding matrices P'_i stays the same, namely lr for each layer. Unfortunately, for small values of r (in particular, $2 \leq r \leq 4$), there is a high probability that two polynomials $\phi_{l,i}, \phi_{l,j}$ for $i \neq j$ have the same coefficients: for $r = 2$, there is only one non-linear coefficient, for $r = 3$, there are only 3, and for $r = 4$, we obtain 6. The corresponding probabilities are therefore $2^{-1}, 2^{-3}$ and 2^{-6}, respectively, that the polynomials $\phi_{l,i}, \phi_{l,j}$ share the same quadratic coefficients. In a linear combination of these two polynomials, the rank of the corresponding matrix will therefore drop by r. This change defeats the `lowRank` algorithm from Figure 4.4 as it only uses the matrix representation of the public key polynomials p_i. That way, it will not only find solutions of the layer l, but also for such linear combinations. To attack RSSE(2)PKC, it is therefore advisable to use the `highRank` algorithm from Figure 4.3 in connection with Dickson's theorem (cf Section 4.4.2).

RSE(2)PKC. The system RSE(2)PKC is a special instance of the previously described RSSE(2)PKC system: the polynomials $\phi_{l,i}$ are required to be step-wise bijections, *i.e.*, the function $(\phi_{l,1}, \ldots, \phi_{l,r}) : \mathbb{F}_2^r \to \mathbb{F}_2^r$ is a bijection for all $l \in \{1, \ldots, L\}$. This way, the whole system $\mathcal{P}$ becomes a bijection and it is possible to recover the solution x step by step without any ambiguity. As being a bijection is a rather strong requirement for a system of multivariate polynomials, the problem described in the previous section becomes more severe as we have far less choices for the coefficients in the quadratic terms. Still, using the high-rank rather than the low-rank attack should overcome this problem.

In [KS04c, Sect. 3.2], the authors suggest $q = 2, r \leq 10$ for their scheme which leads to a maximal value of 2^{10} for q^r. Therefore, we expect all attacks from the previous section to be efficient against these schemes.

Challenges. In [KS04c, Sect. 6], Kasahara and Sakai propose two challenges with the following parameters: $\mathbb{F} = \mathrm{GF}(2)$, $m = n = 100$ and $r = 4, 5$. They gave both the public key and a value $y \in \mathbb{F}^m$ and asked for the corresponding $x \in \mathbb{F}^n$. Using a (highly unoptimised) Magma [MAG] programme, we were able to break this challenge in a few hours on an AMD Athlon XP 2000+. As the challenge files have a size of 128 kByte each, we omitted to include them in this thesis but refer to `http://www.osaka-gu.ac.jp/php/kasahara/publickey.html`. This webpage states both the problem and acknowledges that we computed the required solution.

For our attack, we implemented the inversion attack against the low-rank side (cf sections 4.4.2 and 4.4.3). As pointed out earlier, the attack should have been more efficient using an attack against the high-rank side in combination with Dickson's theorem (cf Section 4.4.2). In particular, we computed the solution x for the given value y. The two solutions are (in vector-notation, starting with x_1 at the left):

- $r = 4$: (0 0 1 1 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 0 0 1 1 0 1 0 1 0 0 1),

- $r = 5$: (1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1).

These results have been confirmed by Kasahara and Sakai [KS04a].

Intermediate Discussion. Apart from the attacks presented in this section, we also want to point out that the generic birthday attack for signature schemes applies against the parameter choice $q = 2$ and $n = m = 100$. This attack is applicable against any signature scheme and uses the fact that we can expect at least one collision after $O(2^{n/2})$ computations from both sides, *i.e.*, changing the message $y \in \mathbb{F}^m$ and the signature $x \in \mathbb{F}^n$. In this case, the workload becomes only $O(2^{50})$ different messages y and $O(2^{50})$ input x. As Kasahara and Sakai do not use special constructions as, *e.g.*, Feistel-Patarin-Networks [CGP01], the generic birthday attack applies in particular against RSE(2)PKC, and RSSE(2)PKC. In addition, the hybrid type construction from the following section is also affected.

Hybrid type construction. In [KS04b, Sect. 4.2], Kasahara and Sakai propose a so-called "hybrid type construction" to enhance the security of RSSE(2)PKC. To simplify this explanation, we restrict to the case with two branches as this is sufficient to point out its vulnerability to the attacks described in this thesis.

In this case, the private polynomials p'_i are partitioned into two sets: the polynomials $p'_1, \ldots, p'_{m/2}$ are constructed as for RSSE(2)PKC (see above). However, the construction of the other polynomials now involves a third type of polynomial, denoted σ. For $L/2 < l \leq L$ and $1 \leq i \leq r$ we have:

$$\begin{aligned} p'_{lr+i}(x') \quad &:= \quad \phi_{l,i}\left(x'_{(l-1)r+1}, \ldots, x'_{lr}\right) + \psi_{l,i}\left(x'_1, \ldots, x'_{(l-1)r}\right) \\ &\qquad + \sigma_{lr+i}\left(x'_1, \ldots, x'_{(L/2)}\right) . \end{aligned}$$

As for $\phi_{l,i}$ and $\psi_{l,i}$, the polynomials σ_{lr+i} are quadratic polynomials with randomly chosen coefficients and no constant term α. All of them depend on the first $L/2$ variables only. Therefore, the overall structure of the private polynomials p'_i in terms of the rank of their matrix representation P'_i does not change and the attacks of this thesis are still applicable.

4.4.6 Discussion

As outlined in Section 3.1.2, regular STS may be generalised by different step-sizes and also different number of equations in each individual level, denoted $r_1, \ldots, r_L \in \mathbb{N}$ and $m_1, \ldots, m_L \in \mathbb{N}$, respectively. Moreover, we may consider these L-tuples as part of the private key; only their sums n and m are public. However, the internal structure of the private key stays the same, in particular, we still obtain the chain of kernels of the private key polynomials. The only part of the attack we have to be careful about are the values r_1 and m_L, *i.e.*, the number of variables in the first layer and the number of equations in the last layer. If the first is too large, the attack at the low-rank side is no longer effective while a high value of the latter may preclude the attack from the high-rank side.

Using gSTS (general Stepwise-Triangular Scheme) for a signature scheme allows us to choose $r_1 \gg m_1$. However, in this case we may not allow $r_L \ll m_L$ as this leads to a highly overdetermined system of equations — which has only $q^{m_L - r_L}$ solutions on average. The situation is reversed for encryption schemes. Here, we may have $r_L \ll m_L$ but not $r_1 \gg m_1$. As the system has a solution for $y = \mathcal{P}(x)$ by construction, a large value of m_L does not provide a problem here. Unfortunately, we are not able to find it back if the value for r_1 and consequently q^{r_1} is too large.

Therefore, gSTS will either fall to an attack from the high-rank or from the low-rank side. In both cases the construction is insecure. We want to point out that gSTS is a generalisation of the Triangular Plus-Minus (TPM) construction.

In particular, we relax the condition that there is only one new variable and one new equation at each intermediate level (cf Sect. 3.1.2).

Highly Overdetermined Schemes. When the scheme has more equations than variables, *i.e.*, for $m > n$, we need to adapt the algorithm `LowRankAttack` (cf Section 4.4.2). Instead of picking one vector in each layer, we need to consider $\lambda := \lceil \frac{m}{n} \rceil$ vectors $v^1, \ldots, v^\lambda \in \mathbb{F}^n$ simultaneously. Now we have to solve the system of equations $\sum_{i=0}^{m} v_i^j (wP_i) = 0$ for $j \in \{1, \ldots, \lambda\}$ in order to have enough information in order to recover the rows of $\tilde{T}$. As for the case $m \leq n$, this system of linear equations has q^{lr} solutions if and only if all vectors $v^1, \ldots, v^\lambda$ are in the kernel $\ker_l$. Consequently, the complexity for the LowRankAttack increases exponentially with λ and is equal to $O(mn^3Lq^{\lambda r})$. In practice we will have small values for λ as highly overdetermined systems of quadratic equations are easy to solve [CGMT02]. This approach for dealing with overdetermined system of equations has earlier been described by Goubin and Courtois [GC00].

All in all, we see that STS without any special trapdoor for the individual layers — in particular the first and the last layer — cannot be securely used. In this context we want to point out that the attacks described in sections 4.4.2 and 4.4.2 work not only for STS but for any schemes which has "close" ranks of its private equations. In [YC04a], they are consequently called "crawling attacks" as it is possible to "crawl" from one layer with specific rank to another. For this crawling, we have a workload of q^Δ where $\Delta \in \mathbb{N}$ denotes the difference in ranks between these two layers. Still, to apply crawling attacks in the context of STS schemes it is usually necessary to get rid of either the first or the last layer of private key equations.

Affine Transformations. Until now, we concentrated on $S, T \in \text{Hom}^{-1}(\mathbb{F}^n) \times \text{Hom}^{-1}(\mathbb{F}^m)$ rather than $S, T \in \text{Aff}^{-1}(\mathbb{F}^n) \times \text{Aff}^{-1}(\mathbb{F}^m)$. We will now investigate why the attacks we developed so far are actually sufficient and cannot be countered by replacing S, T with affine transformations.

Therefore, consider two affine transformations $S \in \text{Aff}^{-1}(\mathbb{F}^n)$, $T \in \text{Aff}^{-1}(\mathbb{F}^m)$. Then there exists a unique, invertible matrix $M_S \in \mathbb{F}^{n \times n}$ (resp. $M_T \in \mathbb{F}^{m \times m}$) and a unique vector $v_s \in \mathbb{F}^n$ (resp. $v_t \in \mathbb{F}^m$) which describe the affine transformation S (resp. T) by $S(x) = M_S x + v_s$ where $x \in \mathbb{F}^n$ is an input vector (resp. $T(x) = M_T x + v_t$ for $x \in \mathbb{F}^m$). Moreover, we can rewrite the affine transformation S as $S(x) = (\overline{x} + v_s) \circ (M_S x)$ where $\overline{x}$ denotes the output of $M_S x$. In addition, we can rewrite the affine transformation T as $T(x) = (M_T \hat{x}) \circ (x + M_T^{-1} v_t)$, where $\hat{x}$ denotes the output of $x + M_T^{-1} v_t$. As M_T is an invertible matrix, the matrix $M_T^{-1} \in \mathbb{F}^{m \times m}$ exists and is unique. We now express the public key as a composition of the private key

$$\begin{aligned}\mathcal{P} &= T \circ \mathcal{P}' \circ S \\ &= [(M_T\hat{x}) \circ (\tilde{x} + M_T^{-1}v_t)] \circ \mathcal{P}' \circ [(\overline{x} + v_s) \circ (M_S x)]\,,\end{aligned}$$

where $\tilde{x}$ is the output of $\mathcal{P}' \circ [(\overline{x} + v_s) \circ (M_S x)]$ and $\hat{x}$ is the output of $(\tilde{x} + M_T^{-1}v_t) \circ \mathcal{P}' \circ [(\overline{x} + v_s) \circ (M_S x)]$. We have

$$\begin{aligned}\mathcal{P} &= (M_T\hat{x}) \circ [(\tilde{x} + M_T^{-1}v_t) \circ \mathcal{P}' \circ (\overline{x} + v_s)] \circ (M_S x) \\ &= (M_T\hat{x}) \circ \mathcal{P}'' \circ (M_S x)\end{aligned}$$

for some system of equations $\mathcal{P}''$. As both $(\overline{x} + v_s)$ and $(\tilde{x} + M_T^{-1}v_t)$ are transformations of degree 1, they do not change the overall degree of $\mathcal{P}''$, *i.e.*, as $\mathcal{P}'$ consists of equations of degree 2 at most, so will $\mathcal{P}''$. In addition, due to its construction, $(M_S, \mathcal{P}'', M_T)$ forms a valid private key for the public key $\mathcal{P}$ and the layer-structure of STS is not affected by these two operations. Therefore, we can "collect" the constant terms of the two affine transformations S, T in the central equations and use the original cryptanalysis against STS with linear rather than affine transformations. We see in Section 5 that similar conclusions can be drawn for other $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes.

4.5 Attacks against MIA

While the author of this thesis has performed substantial work in the cryptanalysis of other schemes, this is not the case for the MIA class. We therefore quote some important attacks. Recall that we have $Y' := X'^{q\lambda+1}$ as the central equation in MIA systems.

The original MIA scheme was broken in [Pat95]. To outline this attack we set $X' := \phi^{-1}(S^{-1}(x))$ and $Y' := \phi^{-1}(T^{-1}(x))$ for $x \in \mathbb{F}^n$. The key observation for his attack are the following equations. First apply the operation $g : A \to A^{q^\theta - 1}$ to both sides of the MIA equation. This yields:

$$Y'^{q^\lambda - 1} = X'^{q^{2\lambda} - 1}\,.$$

Multiplying both sides with $X'Y'$ leads to

$$X'Y'^{q^\theta} = Y'X'^{q^{2\lambda}}\,.$$

We see that the above equation is linear in X' and Y' in terms of the vector space $\mathbb{F}^n$. Starting from this, Patarin shows how to compute equations linear in $x_1, \ldots, x_n$ from the public key alone. Therefore, this attack falls in the class of inversion attacks. A newer attack with the same result but using differential cryptanalysis has been reported in [FGS05].

The main source of cryptanalytic results on MIA and important variations is [PGC98a]. There, the variation MIA- (under the name C^{*--}) is considered to be secure for q^r being larger than some security level $C \in \mathbb{N}$, *e.g.*, $C = 2^{80}$. The reason is an extension of the above cryptanalysis which also works for MIA- instead of MIA+. Interestingly, the plus modification does not have any impact on the above crytanalysis, *i.e.*, MIA+ is not at all more secure than MIA itself. Indeed, all the equations needed for the above inversion attack still exist in MIA+: hence, we can still compute them.

There has also been an attack against an early version of Sflash [GM02], mainly using the fact that this early version used coefficients from GF(2) rather than from GF(128) for the private key transformations $S, T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$. This way, an exhaustive search over one column of S was possible. This is another example for the shortcomings of the "/" modification from Section 3.2.3.

In addition, there is the attack [GSB01] which shows how to recover the vectors $v_S, v_T \in \mathbb{F}^n$ of the affine transformations $S, T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$. Although this result cannot be extended to the matrices $M_S, M_T \in \mathbb{F}^{n \times n}$ of these transformations, it shows that the use of affine instead of linear transformations S, T does not increase the overall security of schemes of the MIA type.

As a more recent development is the scheme MIAi, which has been developed in [Din04], cf Section 3.2.8. There, it is called "Perturbed Matsumoto-Imai" (PMI). It has been broken in [FGS05], using a differential attack to distinguish between noisy and non-noisy parts of the message space $\mathbb{F}^n$. Using only the non-noisy parts, it is possible to launch the original cryptanalysis of Patarin against MIA, or the new attack developed in [FGS05]. In terms of running time, this new attack needs $O(q^{3w})$ operations for w being the perturbation dimension. Recalling that the workload of the legitimate user is also proportional to q^w we see that q^w needs to be small for efficient schemes. Hence, MIAi has to be considered broken.

So at present, we see that MIA- is the most time-efficient secure version of a signature scheme based on multivariate quadratic equations. As we will see later, the main problem for its use in practice is the size of the public key.

4.6 Attacks against HFE

In this section, we give a brief overview of recent attacks against HFE. For a more detailed but partly outdated analysis, we refer to [Pat96b]. A newer analysis can be found in [WP04], which also forms the basis for this section.

4.6.1 Kipnis-Shamir: Recover the Private Key

In [KS99], Kipnis and Shamir show how to recover the private key of HFE from the system of public equations. The key point of this attack is to express the private key (*i.e.*, polynomials over the finite field $\mathbb{F}$) as sparse univariate polynomials over the extension field $\mathbb{E}$. We refer to Section 2.4 for the idea of expressing the private key over the extension field $\mathbb{E}$. In addition, they observe that the special choice of the private polynomial P in HFE gives rise to a matrix equation with very small rank (*e.g.*, rank 13 for a 100×100 matrix). In [Cou01, Sect. 8], their attack is improved and has now a workload of $\binom{n}{\mathrm{Rank}P}^{\omega} = \mathcal{O}\left(n^{\log_q d + \mathcal{O}(1)}\right)$. In this formula, Rank$P$ is the rank of the private polynomial P in matrix form over the extension field $\mathbb{E}$, d its degree as a polynomial over $\mathbb{E}$, and $\omega \approx 2.7$ the expected workload to solve linear equations. The attack is only applicable against basic HFE, *i.e.*, it fails for all its variations. On the other hand, it is the only attack known so far which can recover the private key of HFE.

In their paper, Kipnis and Shamir also introduce the "reliniarization" technique which can solve quadratic equations with about $0.1n^2$ linearly independent equations in n variables. For the traditional linearisation technique, we need about $0.5n^2$ many equations. This technique has been improved in [CKPS00]. Still, as shown in [AFI$^+$04], XL is always slower than the algorithms from Faugère.

4.6.2 Faugère: Fast Gröbner Bases

In 2002, Faugère reported to have broken the HFE-Challenge I in 96 hours, using an AlphaServer DS20E (EV68 833 Mhz) with 4 GByte of RAM [Fau02]. In this challenge, we have $q = 2$, $n = 80$, and a degree of $d = 96$ for the central equations. The attacker is given the public key and also a value $y \in \mathbb{F}^m$. The task is to compute at least one $x \in \mathbb{F}^n$ which matches the given y. Interestingly, Faugère was able to compute four different solutions. This highlights the fact that HFE is *not* injective. Since then, his attacks have been improved and in 2003, Faugère and Joux published joint work on the security of HFE [FJ03] (cf [Fau03] for a more technical version). In a nutshell, their attack uses a fast algorithm to compute the Gröbner basis of a system of polynomial equations. By theoretical and empirical studies they show that inverting basic HFE given the public key alone is polynomial for a fixed degree d in the private key polynomial $P \in \mathbb{E}[X]$. The attack-complexity for different degrees is shown in Table 4.1

For HFEv, Faugère and Joux outline in [FJ03, Sect. 4.1] that the cryptanalysis is not more difficult in this case. But for HFE-, they get a higher workload. For the original Quartz-scheme, they establish a workload of $\approx 2^{62}$ — exploiting some further properties of their attack. However, these additional improvements are not within the scope of this thesis. Unfortunately, they only give the number of matrix operations — which are difficult to relate to number of 3-DES

Table 4.1: Attack complexity against basic HFE for different degrees d

Degree d	$16 < d \leq 128$	$128 < d \leq 512$	$512 > d$
Attack (asymptotical)	$\mathcal{O}(n^8)$	$\mathcal{O}(n^{10})$	$\geq \mathcal{O}(n^{12})$
Attack (for $n = 103$)	$\approx 2^{54}$	$\approx 2^{66}$	$\approx 2^{80}$

computations. Hence, we could not compute the overall security level.

Using the estimations of [FJ03, Sect. 4.1, 5.2–5.4] on Quartz, we establish that a degree of 129 and 7 equations removed (thus, without the modification HFEv) has an attack complexity of $\approx 2^{86}$. The corresponding "Quartz-7m"-scheme is therefore secure again. In fact, a similar result has been achieved 2002 in [CDF03] by increasing the degree d of the private polynomial to 257. However, this estimation was only based on [Fau02]. In the light of the article [FJ03] it turns out to be inaccurate.

4.6.3 Variations of HFE

Hidden field equations were mainly used with the minus and the "v" modification so far. The cryptanalysis of [KS99] becomes ineffective if any variation is applied. However, the later work of Faugère and Joux [FJ03] proves very efficient against HFE, HFE+, and also to some extent against HFEv. Still, the method HFE- proves a very efficient way to counter this attack.

Recently, Ding and Schmidt suggested to use the variation HFEi — they called it IPHFE ("Internal Perturbation of HFE", cf Section 3.2.8). Unfortunately, there is not much independent research known about strength of this new scheme. However, we expect it to be secure against Gröbner attacks. Given that MIAi (see above) has been broken rather unexpectedly, we suggest to wait some time before using HFEi in applications.

4.7 Discussion

In this chapter, we gave an overview on cryptanalytic results for the four basic classes UOV, STS, MIA, and HFE plus some of their variations. As we saw there, only UOV with well-chosen parameters withstands all attacks while the basic classes of the other schemes are vulnerable to attacks.

At present, MIA-, HFE-, and UOV seem to be mature enough to be used in practical applications. Interestingly, all of them are only useful in the context of signature schemes.

For all other designs, it is simply too early to use them. This is in particular true for the newly developed "i" modification: while it would allow an encryption scheme — either based on MIA or HFE — the MIAi variation has been broken, and the HFEi version is only known since a year.

For STS, the situation is worse: the basic scheme cannot be used at all for secure encryption or even signature schemes. The reason is very efficient attacks against the first and the last layer of this multi-layer scheme. Interestingly, we will see in Section 6.2 that the STS class is actually used in connection with other schemes to construct secure signature schemes. Until now, the STS class could not be used either for encryption schemes.

Therefore, we have to conclude that the construction of encryption schemes based on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations is a difficult task. However, not having a way of exchanging secrets, *e.g.*, session keys, is a serious obstacle for any public key scheme. It would therefore be very interesting to obtain an encryption scheme based on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials.

Chapter 5

Equivalent Keys

After dealing with cryptanalytic results in the previous chapter, we now move on to a different subject: the question of equivalent private keys for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes. A graphical representation of this idea is presented in Figure 5.1. At first glance, this question seems to be purely theoretical. But for

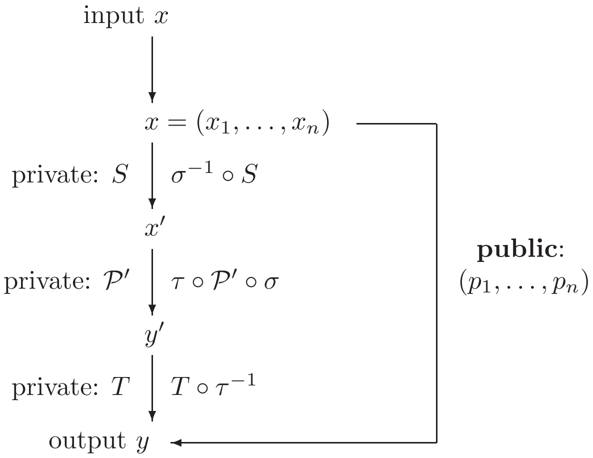

Figure 5.1: Equivalent private keys using affine transformations σ, τ

practical applications, we need memory and time efficient instances of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. One important point in this context is the overall *size* of the private key: in restricted environments such as smart cards, we want it as small as possible. Hence, if we can show that a given private key is only a representative of a much larger class of equivalent private keys, it makes sense to compute (and store) only a normal form of this key. Similar, we should construct new $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes such that they do not have a large number of equivalent private keys but only a small number, preferable only one, per equivalence class. This way, we make optimal use of the randomness in the private key space and neither waste computation time nor storage space without any security benefit.

This chapter is based on two papers [WP05c] and [WP05b]. The article [WP05c] is the first publicly available, systematic treatment of the question of equivalent keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems. So our own contribution consists of all but one of the sustaining transformations described here and their application to the MIA, MIA-, HFE-, HFEv, HFEv-, UOV, and the STS class. For the HFE class, the application of the so-called "additive sustainer" was known previously. Still, it was not recognised before that it could be applied also to other $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems such as UOV. The theorem for the STS class has not been published before.

We want to mention that the question of equivalent keys for the MIA class has been independently studied by Prof. Dobbertin. We learned about his results after the paper [WP05c] has been accepted for publication at PKC 2005. Due to his suggestion, we added Section 5.4 in which we show that the sustaining transformations given in Theorem 5.3.7 are actually *all* sustaining transformations possible, using some reasonable assumptions. In addition, we also extended his idea to the MIO class, cf Section 6.4.1.

5.1 Initial Considerations

Before discussing concrete schemes, we start with some general observations and definitions. Obviously, the most important term in this chapter is "equivalent private keys", see Figure 5.1 for a graphical representation. We start by properly introducing it:

Definition 5.1.1 *We call two private keys*

$$(T, \mathcal{P}', S), (\tilde{T}, \tilde{\mathcal{P}'}, \tilde{S}) \in \mathit{Aff}^{-1}(\mathbb{F}^m) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^n)$$

"equivalent" if they lead to the same public key, i.e.*, if we have*

$$T \circ \mathcal{P}' \circ S = \mathcal{P} = \tilde{T} \circ \tilde{\mathcal{P}'} \circ \tilde{S}\,.$$

In order to find equivalent keys, we consider the following transformations:

DEFINITION 5.1.2 *Let* $(S, \mathcal{P}', T) \in \mathit{Aff}^{-1}(\mathbb{F}^m) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^n)$*, and consider the four transformations* $\sigma, \sigma^{-1} \in \mathit{Aff}^{-1}(\mathbb{F}^n)$ *and* $\tau, \tau^{-1} \in \mathit{Aff}^{-1}(\mathbb{F}^m)$*. Moreover, let*

$$\mathcal{P} = T \circ \tau^{-1} \circ \tau \circ \mathcal{P}' \circ \sigma \circ \sigma^{-1} \circ S\,. \tag{5.1}$$

We call the pair $(\sigma, \tau) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathit{Aff}^{-1}(\mathbb{F}^m)$ *"sustaining transformations" for an* $\mathcal{MQ}$*-system if the "shape" of* $\mathcal{P}'$ *is invariant under the transformations* σ *and* τ*. For short, we write* $(\sigma, \tau) \bullet (S, \mathcal{P}', T)$ *for (5.1.2) and* (σ, τ) *sustaining transformations (cf Figure 5.1).*

Remark 5.1.3 *In the above definition, the meaning of "shape" is still open. In fact, its meaning has to be defined for each* $\mathcal{MQ}$*-system individually. For example, in HFE (cf Section 3.1.4), it is the bounding degree* $d \in \mathbb{N}$ *of the polynomial* $P'(X')$*. In the case of MIA, the "shape" is the fact that we have a single monomial with factor 1 as the central equation (cf Section 3.1.3). In general and for* σ, τ *sustaining transformations, we are now able to produce equivalent keys for a given private key by* $(\sigma, \tau) \bullet (S, \mathcal{P}', T)$*. A trivial example of sustaining transformations is the identity transformation,* i.e.*, to set* $\sigma = \tau = id$*.*

Lemma 5.1.4 *Let* $\sigma \in \mathit{Aff}^{-1}(\mathbb{F}^n), \tau \in \mathit{Aff}^{-1}(\mathbb{F}^m)$ *be sustaining transformations. If the two structures* $G := (\sigma, \circ)$ *and* $H := (\tau, \circ)$ *form a subgroup of the affine transformations, they produce equivalence relations within the private key space.*

PROOF. We start with a proof of this statement for $G := (\sigma, \circ)$. First, we have reflexivity as the identity transformation is contained in G. Second, we have symmetry as subgroups are closed under inversion. Third, we also have transitivity as subgroups are closed under composition. Therefore, the subgroup G partitions the private key space into equivalence classes. The proof for the subgroup $H := (\tau, \circ)$ is analogous. □

Remark 5.1.5 *We want to point out that the above proof does not use special properties of sustaining transformations, but the fact that these are a subgroup of the group of affine transformations. Hence, the proof does not depend on the term "shape" and is therefore valid even if the latter is not rigorously defined yet. In any case, instead of proving that sustaining transformations form a subgroup of the affine transformations, we can also consider normal forms of private keys. As we see below, normal forms have some advantages to avoid double counts in the private key space.*

5.2 Sustaining Transformations

In this section, we discuss several examples of sustaining transformations. In addition, we consider their effect on the central transformation $\mathcal{P}'$.

5.2.1 Additive Sustainer

For $n = m$, let $\sigma(X) := (X + A)$ and $\tau(X) := (X + A')$ for some elements $A, A' \in \mathbb{E}$. As long as the transformations σ, τ keep the shape of the central equations $\mathcal{P}'$ invariant, they form sustaining transformations.

In particular, we are able to change the constant parts $v_s, v_t \in \mathbb{F}^n$ or $V_S, V_T \in \mathbb{E}$ of the two affine transformations $S, T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ to zero, *i.e.*, to obtain a new key $(\hat{S}, \hat{\mathcal{P}}', \hat{T})$ with $\hat{S}, \hat{T} \in \mathrm{Hom}^{-1}(\mathbb{F}^n)$.

Remark 5.2.1 *This is a very useful result for cryptanalysis as it allows us to "collect" the constant terms in the central equations $\mathcal{P}'$. For cryptanalytic purposes, we therefore only need to consider the case of linear transformations $S, T \in Hom^{-1}(\mathbb{F}^n)$.*

The additive sustainer also works if we interpret it over the vector space $\mathbb{F}^n$ rather than the extension field $\mathbb{E}$. To distinguish this case clearly from the setting above, we write $a \in \mathbb{F}^n, a' \in \mathbb{F}^m$ here. In particular, we can also handle the case $n \neq m$ now. However, in this case it may happen that we have $a' \in \mathbb{F}^m$ and consequently $\tau : \mathbb{F}^m \to \mathbb{F}^m$. Nevertheless, we can still collect all constant terms in the central equations $\mathcal{P}'$.

If we look at the central equations as multivariate polynomials, the additive sustainer will affect the constants α_i and $\beta_{i,j} \in \mathbb{F}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. A similar observation is true for central equations over the extension field $\mathbb{E}$: in this case, the additive sustainer affects the additive constant $A \in \mathbb{E}$ and the linear factors $B_i \in \mathbb{E}$ for $0 \leq i < n$.

5.2.2 Big Sustainer

We now consider multiplication in the (big) extension field $\mathbb{E}$, *i.e.*, we have $\sigma(X) := (BX)$ and $\tau(X) := (B'X)$ for $B, B' \in \mathbb{E}^*$. Again, we obtain a sustaining transformation if this operation does not modify the shape of the central equations as $(BX), (B'X) \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$.

The big sustainer is useful if we consider schemes defined over extension fields as it does not affect the overall degree of the central equations over this extension field.

5.2.3 Small Sustainer

We now consider vector-matrix multiplication over the (small) ground field $\mathbb{F}$, *i.e.*, we have $\sigma(x) := Diag(b_1, \ldots, b_n)x$ and $\tau(x) := Diag(b'_1, \ldots, b'_m)x$ for the non-zero coefficients $b_1, \ldots, b_n, b'_1, \ldots, b'_m \in \mathbb{F}^*$ and $Diag(b)$, $Diag(b')$ the diagonal matrices on both vectors $b \in \mathbb{F}^n$ and $b' \in \mathbb{F}^m$, respectively.

In contrast to the big sustainer, the small sustainer is useful if we consider schemes which define the central equations over the ground field $\mathbb{F}$ as it only introduces a scalar factor in the polynomials $(p'_1, \ldots, p'_m)$.

5.2.4 Permutation Sustainer

For the transformation σ, this sustainer permutes input-variables of the central equations while for the transformation τ, it permutes the polynomials of the central equations themselves. As each permutation has a corresponding, invertible permutation-matrix, both $\sigma \in S_n$ and $\tau \in S_m$ are also affine transformations. The effect of the central equations is limited to a permutation of these equations and their input variables, respectively.

5.2.5 Gauss Sustainer

Here, we consider Gauss operations on matrices, *i.e.*, row and column permutations, multiplication of rows and columns by scalars from the ground field $\mathbb{F}$, and the addition of two rows/columns. As all these operations can be performed by invertible matrices; they form a subgroup of the affine transformations and are hence a candidate for a sustaining transformation.

The effect of the Gauss sustainer is similar to the permutation sustainer and the small sustainer. In addition, it allows the addition of multivariate quadratic polynomials. This will not affect the shape of some $\mathcal{MQ}$-schemes.

5.2.6 Frobenius Sustainer

Definition 5.2.2 *Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements and $\mathbb{E}$ its n-dimensional extension. Moreover, let $H := \{i \in \mathbb{Z} : 0 \leq i < n\}$. For $a, b \in H$ we call $\sigma(X) := X^{q^a}$ and $\tau(X) := X^{q^b}$* Frobenius transformations *(cf Lemma 2.1.5).*

Obviously, Frobenius transformations are linear transformations with respect to the ground field $\mathbb{F}$. The following lemma establishes that they also form a group:

Lemma 5.2.3 *Frobenius transformations are a subgroup in $Hom^{-1}(\mathbb{F}^n)$.*

Proof. First, Frobenius transformations are linear transformations, so associativity is inherited from them. Second, the set H from Definition 5.2.2 is not empty for any given $\mathbb{F}$ and $n \in \mathbb{Z}^+$. Hence, the corresponding set of Frobenius transformations is not empty either. In particular, we notice that the Frobenius transformation X^{q^0} is the neutral element of this group.

In addition, the inverse of a Frobenius transformation is also a Frobenius transformation: Let $\sigma(X) := X^{q^a}$ for some $a \in H$. Working in the multiplicative group $\mathbb{E}^*$ we observe that we need $q^a \cdot A' \equiv 1 \pmod{q^n - 1}$ for $A' \in \mathbb{N}$ to obtain the inverse function of σ. We notice that $A' := q^{a'}$ for $a' := n - a \pmod{n}$ yields the required and moreover $\sigma^{-1} := X^{q^{a'}}$ is a Frobenius transformation as $a' \in H$.

So all left to show is that for any given Frobenius transformations σ, τ, the composition $\sigma \circ \tau$ is also a Frobenius transformation, *i.e.*, that we have closure.

Let $\sigma(X) := X^{q^a}$ and $\tau(X) := X^{q^b}$ for some $a, b \in H$. So we can write $\sigma(X) \circ \tau(X) = X^{q^{a+b}}$. If $a + b < n$ we are done. Otherwise $n \leq a + b < 2n$, so we can write $q^{a+b} = q^{n+s}$ for some $s \in H$. Again, working in the multiplicative group E^* yields $q^{n+s} \equiv q^s \pmod{q^n - 1}$ and hence, we established that $\sigma \circ \tau$ is also a Frobenius transformation. This completes the proof that all Frobenius transformations form a group. □

Frobenius transformations usually change the degree of the central equation $\mathcal{P}'$. But taking $\tau := \sigma^{-1}$ cancels this effect and hence preserves the degree of $\mathcal{P}'$. Therefore, we can speak of a Frobenius sustainer (σ, τ). So there are n Frobenius sustainers for a given extension field $\mathbb{E}$.

It is tempting to extend this result to the case of powers of the characteristic of $\mathbb{F}$. However, this is not possible as $x^{\mathrm{char}\mathbb{F}}$ is not a linear transformation in $\mathbb{F}$ for $q \neq p$ where p denotes the characteristic of the finite field $\mathbb{F}$ and $q := |\mathbb{F}|$ the number of its elements.

Remark 5.2.4 *All six sustainers presented so far form groups and hence partition the private key space into equivalence classes (cf Lemma 5.1.4).*

5.2.7 Reduction Sustainer

The reduction sustainer is quite different from the transformations studied so far. The main reason is that it is not applied to a basic form of an $\mathcal{MQ}$-trapdoor, but to an $\mathcal{MQ}$-trapdoor which uses the minus (or reduction) modification (cf Section 3.2.1). We recall that we have $\mathcal{P} := R \circ T \circ \mathcal{P}' \circ S$ here where $R : \mathbb{F}^n \to \mathbb{F}^{n-r}$ denotes a *reduction* or *projection* (see Section 2.4 for details). In addition, we have $S, T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n)$. More formally, we consider the function $R(x_1, \ldots, x_n) := (x_1, \ldots, x_{n-r})$, *i.e.*, we neglect the last r components of the vector $(x_1, \ldots, x_n)$. Although this modification looks rather easy, we recall from Chapter 4 that it proves powerful to defeat a wide class of cryptographic attacks against several $\mathcal{MQ}$-schemes, including HFE and MIA, *e.g.*, the attack introduced in [FJ03].

For the corresponding sustainer, we consider the affine transformation T in matrix representation, *i.e.*, we have $T(x) := Mx + v$ for some invertible matrix $M \in \mathbb{F}^{m\times m}$ and a vector $v \in \mathbb{F}^m$. We observe that any change in the last r columns of M or v does not affect the result of R (and hence $\mathcal{P}$). Hence, we can choose these last r columns without affecting the public key. Inspecting Lemma 2.2.2, we see that this gives us a total of

$$q^r \prod_{i=n-r-1}^{n-1} \left(q^n - q^i\right)$$

choices for v and M, respectively, that do not affect the public key equations $\mathcal{P}$.

When applying the reduction sustainer together with other sustainers, we have to make sure that we do not count the same transformation twice, cf the corresponding proofs in the following section.

5.3 Application to $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Schemes

All necessary tools at hand, we show how to apply suitable sustaining transformations to the multivariate schemes. We want to stress that the reductions in size we achieve in this section represent lower rather than upper bounds: additional sustaining transformations can further reduce the key space of these schemes. The only exception for this rule is the MIA class: due to the tightness proof in Section 5.4, we know that only the big sustainer and the Frobenius sustainer can be applied here. Unfortunately, the details of this tightness proof are cumbersome and we do not see how it can be extended in a straightforward way to the other schemes discussed in this section.

5.3.1 Hidden Field Equations

We start with the HFE class (cf Section 3.1.4 for its definition) as the overall proof ideas can be demonstrated most clearly here. In fact, we will use some of these ideas again for the MIA class.

We recall that the central equation $Y' = P(X')$ of HFE is represented over the extension field $\mathbb{E}$ and that the degree of the polynomial P is bounded by d. We need this condition to allow efficient inversion of the equation $P(X') = Y'$ for given $Y' \in \mathbb{E}$. So the *shape* of HFE is in particular this degree d of the private polynomial P. Moreover, we observe that there are no restrictions on its coefficients $C'_{i,j}, B'_k, A' \in \mathbb{E}$ for $i, j, k \in \mathbb{N}_0$ and $q^i, q^i + q^j \leq d$. Hence, we can apply both the additive and the big sustainer (cf sect. 5.2.1 and 5.2.2) without changing the shape of this central equation.

Theorem 5.3.1 *For $K := (S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^n)$ a private key in HFE, we have*

$$n.q^{2n}(q^n - 1)^2$$

equivalent keys.

Proof. To prove this theorem, we consider normal forms of private keys: let $\tilde{S} \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ being the affine transformation we start with. First we compute $\hat{S}(X) := \tilde{S}(X) - \tilde{S}(0)$, *i.e.*, we apply the additive sustainer. Obviously, we have $\hat{S}(0) = 0$ after this transformation and hence a special fix-point. Second we define $\overline{S}(X) := \hat{S}(X).\hat{S}(1)^{-1}$, *i.e.*, we apply the big sustainer. As the transformation $\hat{S} : \mathbb{E} \to \mathbb{E}$ is a bijection and we have $\hat{S}(0) = 0$, we know that $\hat{S}(1)$ must be non-zero. Hence, we have $\overline{S}(1) = 1$, *i.e.*, we add a new fix-point but still keep the old fix-point as we have $\overline{S}(0) = \hat{S}(0) = 0$. Similar we can compute an affine transformation $\overline{T}(X)$ with $\overline{T}(0) = 0$ and $\overline{T}(1) = 1$ as a normal form of the affine transformation $\tilde{T} \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$. Note that both the additive sustainer and the big sustainer keep the degree of the central polynomial $P(X)$ so we can apply both sustainers on both sides without changing the "shape" of $P(X)$.

Applying the Frobenius sustainer is a little more technical. First we observe that this sustainer keeps the fix-points $\overline{S}(0) = \overline{T}(0) = 0$ and $\overline{S}(1) = \overline{T}(1) = 1$ so we are sure we still deal with equivalence classes, *i.e.*, each given private key has a unique normal form, even with the Frobenius sustainer applied. Now we pick an element $C \in \mathbb{E}\backslash\{0, 1\}$ for which $g := \overline{S}(C)$ is a generator of $\mathbb{E}^*$, *i.e.*, we have $\mathbb{E}^* = \{g^i \mid 0 \leq i < q^n\}$. As $\mathbb{E}$ is a finite field we know that such a generator g exists. Given that $\overline{S}$ is surjective we know that we can find the corresponding $C \in \mathbb{E}\backslash\{0, 1\}$. Now we compute $g_i := \overline{S}(C)^{q^i}$ for $0 \leq i < n$. Using any total ordering "<", we obtain $g_c := \min\{g_0, \ldots, g_{n-1}\}$ for some $c \in \mathbb{N}$ as the smallest element of this set. One example of such a total ordering would be to use a bijection between the sets $\mathbb{E} \leftrightarrow \{0, \ldots, q^n - 1\}$ and then exploiting the ordering of the natural numbers to derive an ordering on the elements of the extension field $\mathbb{E}$. Finally, we define $S(X) := [\overline{S}(X)]^{q^c}$ as new affine transformation. To cancel the effect of the Frobenius sustainer, we define $T(X) := [\overline{T}(X)]^{q^{n-c}}$.

Hence, we have now computed a unique normal form for a given private key. Moreover, we can "reverse" these computations and derive an equivalence class of size $n.q^{2n}.(q^n - 1)^2$ this way as we have

$$(BX^{q^c} + A, B'X^{q^{n-c}} + A') \bullet (S, \mathcal{P}', T) \text{ for } B, B' \in \mathbb{E}^*, A, A' \in \mathbb{E} \text{ and } 0 \leq c < n\,.$$

□

Remark 5.3.2 *To the knowledge of the author, the additive sustainer for HFE has first been reported in [Tol03]; it was used there for reducing the affine transformations to linear ones. In addition, a weaker version of the above theorem can be found in [WP05c].*

For $q = 2$ and $n = 80$, the number of equivalent keys per private key is $\approx 2^{326}$. In comparison, the number of choices for S and T is $\approx 2^{12,056}$. This special choice of parameters has been used in HFE Challenge 1 [Pat96b].

HFE-

We recall that HFE- is the original HFE-class with the minus modification (cf sections 3.2.1 and 5.2.7). In particular, this means that the "shape" of the central polynomial $P'(X')$ is still the same, *i.e.*, all considerations from the previous theorem also apply to HFE-.

Theorem 5.3.3 *For $K := (S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^n)$ a private key in HFE and a reduction parameter $r \in \mathbb{N}$ we have*

$$n.q^{2n}(q^n - 1)(q^{n-r} - 1) \prod_{i=n-r-1}^{n-1} (q^n - q^i)$$

equivalent keys. Hence, the key-space of HFE- can be reduced by this number.

Proof. This proof uses the same ideas as the proof of Theorem. 5.3.1 to obtain a normal form of the affine transformation S, *i.e.*, applying the additive sustainer, the big sustainer and the Frobenius sustainer on this side. Hence, we have a reduction by $n.q^n(q^n - 1)$ keys here.

For the affine transformation T, we also have to take the reduction sustainer into account: we use $\tilde{T}(X) : \mathbb{F}^n \to \mathbb{F}^{n-r}$ and fix $\tilde{T}(0) = 0$ by applying the additive sustainer and $\tilde{T}(1) = 1$ by applying the big sustainer, which gives us q^{n-r} and $q^{n-r} - 1$ choices, respectively. To avoid double counting with the reduction sustainer, all computations were performed in $\tilde{\mathbb{E}} := \mathrm{GF}(q^{n-r})$ rather than $\mathbb{E}$. Again, we can compute a normal form for a given private key and reverse these computations to obtain the full equivalence class for any given private key in normal form. Moreover, we observe that the resulting transformation $\tilde{T}$ allows for $q^r \prod_{i=n-r-1}^{n-1}(q^n - q^i)$ choices for the original transformation $T : \mathbb{F}^n \to \mathbb{F}^n$ without affecting the output of $\tilde{T}$ and hence, keeping the two fix points $\tilde{T}(0) = 0$ and $\tilde{T}(1) = 1$. Therefore, there are a total of $q^{n-r} \cdot q^r \cdot (q^{n-r} - 1) \cdot \prod_{i=n-r-1}^{n-1}(q^n - q^i)$ possibilities for the transformation T without changing the public key equations. Multiplying out the intermediate results for S and T yields the theorem. $\square$

For $q = 2, r = 7$ and $n = 107$, the number of equivalent keys for each private key is $\approx 2^{2129}$. In comparison, the number of choices for S and T is $\approx 2^{23,108}$. This special choice of parameters has been used in the repaired version Quartz-7m of Quartz [CGP01, WP04].

HFEv

Another important variation of Hidden Field Equations is HFEv. In particular, it was used in the signature scheme Quartz (cf Section 6.1.2). It is due to [KPG99] (see Section 3.2.7 for an outline of this idea). As we only considered the general form of the vinegar modification in this section, we now outline the special case of HFEv, as the details of the central polynomial are crucial for understanding Theorem 5.3.5.

Definition 5.3.4 *Let $\mathbb{E}$ be a finite field with degree n' over $\mathbb{F}$, $v \in \mathbb{N}$ the number of vinegar variables, and $P(X)$ a polynomial over $\mathbb{E}$. Moreover, let $(z_1, \ldots, z_v) := s_{n-v+1}(x_1, \ldots, x_n), \ldots, s_n(x_1, \ldots, x_n)$ for s_i the polynomials of $S(x)$ in multivariate representation and $X' := \phi^{-1}(x'_1, \ldots, x'_{n'})$, using the canonical bijection $\phi^{-1} : \mathbb{F}^n \to \mathbb{E}$ and $x'_i := s_i(x_1, \ldots, x_n)$ for $1 \le i \le n'$ as hidden variables. Then define the central equation as*

$$P'_{z_1,\ldots,z_v}(X') := \sum_{\substack{0\le i,j\le d\\ q^i+q^j\le d}} C_{i,j} X'^{q^i+q^j} + \sum_{\substack{0\le k\le d\\ q^k\le d}} B_k(z_1,\ldots,z_v) X'^{q^k} + A(z_1,\ldots,z_v)$$

$$\text{where} \begin{cases} C_{i,j} X'^{q^i+q^j} & \text{for } C_{i,j} \in \mathbb{E} \text{ are the quadratic terms,} \\ B_k(z_1,\ldots,z_v) X'^{q^k} & \text{for } B_k(z_1,\ldots,z_v) \text{ depending linearly on } z_1,\ldots,z_v \text{ and} \\ A(z_1,\ldots,z_v) & \text{for } A(z_1,\ldots,z_v) \text{ depending quadratically on } z_1,\ldots,z_v \end{cases}$$

and a degree $d \in \mathbb{N}$, we say the central equations $\mathcal{P}'$ are in HFEv-shape.

The condition that the $B_k(z_1, \ldots, z_v)$ are affine functions (*i.e.*, of degree 1 in the z_i at most) and $A(z_1, \ldots, z_v)$ is a quadratic function over $\mathbb{F}$ ensures that the public key is still quadratic over $\mathbb{F}$ (cf Section 2.4).

Theorem 5.3.5 *For $K := (S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^m)$ a private key in HFEv, $v \in \mathbb{N}$ the number of vinegar variables, $\mathbb{E}$ an n'-dimensional extension of $\mathbb{F}$ where $n' := n - v = m$ we have*

$$n' q^{n+n'+vm}(q^{n'} - 1)^2 \prod_{i=0}^{v-1} (q^v - q^i)$$

equivalent keys. Hence, the key-space of HFEv can be reduced by this number.

PROOF. In contrast to HFE-, the difficulty now lies in the computation of a normal form for the affine transformation S rather than the affine transformation T. For the latter, we can still apply the big sustainer and the additive sustainer and obtain a total of $q^m \cdot (q^m - 1) = q^{n'} \cdot (q^{n'} - 1)$ equivalent keys for a given transformation T. Moreover, the HFEv modification does not change the "absorbing behaviour" of the central polynomial P and hence, the proof from Theorem. 5.3.1 is still applicable.

Instead, we have to concentrate on the affine transformation S here. In order to simplify the following argument, we apply the additive sustainer on S and obtain a linear transformation. This reduces the key-space by q^n. In order to make sure that we do not count the same linear transformation twice, we consider a normal form for the now (linear) transformation S

$$\begin{pmatrix} E_m & F_v^m \\ 0 & I_v \end{pmatrix} \text{ with } E_m \in \mathbb{F}^{m \times m}, F_v^m \in \mathbb{F}^{m \times v}\,.$$

In the above definition, we also have I_v the identity matrix in $\mathbb{F}^{v \times v}$. Moreover, the left-lower corner is the all-zero matrix in $\mathbb{F}^{v \times m}$. The reason for this non-symmetry: we may not introduce vinegar variables in the set of oil variables, but due to the form of the vinegar equations, we can introduce oil variables in the set of vinegar variables. This is done by the following matrix. In particular, for each invertible matrix M_S, we have a unique matrix

$$\begin{pmatrix} I_m & 0 \\ G_m^v & H_v \end{pmatrix} \text{ with an invertible matrix } H_v \in \mathbb{F}^{v \times v}.$$

which transfers M_S to the normal form from above. Again, I_m is an identity matrix in $\mathbb{F}^{m \times m}$. Moreover, we have some matrix $G_m^v \in \mathbb{F}^{v \times m}$. This way, we obtain $q^{vm} \prod_{i=0}^{v-1}(q^v - q^i)$ equivalent keys in the "v" modification alone. As stated previously, the identity matrix I_m ensures that the input of the HFE component is unaltered. However, we do not have such a restriction on the input of the vinegar part and can hence introduce the two matrices G_m^v and H_v: they are "absorbed" into the random terms of the vinegar polynomials $B_k(z_1, \ldots, z_v)$ and $A(z_1, \ldots, z_v)$.

For the HFE component over $\mathbb{E}$, we can now apply the big sustainer to S and obtain a factor of $(q^{n'} - 1)$. In addition, we apply the Frobenius sustainer to the HFE component, which yields an additional factor of n'. Note that the Frobenius sustainer can be applied both to S and T, and hence, we can make sure that it cancels out and does not affect the degree of the central polynomial $P_{z_1,\ldots,z_v}(X)$. Again, we can reverse all computations and therefore, obtain equivalence classes of equal size for each given private key in normal form. $\square$

For the case $q = 2, v = 7$ and $n = 107$, the number of equivalent keys for each private is $\approx 2^{1160}$. In comparison, the number of choices for S and T is $\approx 2^{21,652}$.

HFEv-

Here, we combine both the HFEv and the HFE- modification to obtain HFEv-. In fact, the original Quartz scheme (cf Section 6.1.2) was of this type.

Theorem 5.3.6 *For $K := (S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^{m+v}, \mathbb{F}^{m+r})$ a private key in HFEv, $v \in \mathbb{N}$ vinegar variables, a reduction parameter $r \in \mathbb{N}$ and $\mathbb{E}$ an n'-dimensional extension of $\mathbb{F}$ where $n' := n - v$ and $n' = m + r$ we have*

$$n' q^{r+2n'+vn'} (q^{n'} - 1)^2 \prod_{i=0}^{v-1} (q^v - q^i) \prod_{i=n'-r-1}^{n'-1} (q^{n'} - q^i)$$

equivalent keys. Hence, the key-space of HFEv can be reduced by this number.

Proof. This proof is a combination of the two cases HFEv and HFE-. Given that the difficulty for the HFE- modification was in the T-transformation while the difficulty of HFEv was in the S-transformation, we can safely combine the known sustainers without any double-counting. □

For the case $q = 2, r = 3, v = 4$ and $n = 107, n' = 103$, the number of redundant keys is $\approx 2^{1258}$. In comparison, the number of choices for S and T is $\approx 2^{22,261}$. This special choice of parameters has been used in the original version of Quartz [CGP01], as submitted to NESSIE [NES].

5.3.2 Matsumoto-Imai Scheme A

As HFE, the MIA class uses a finite field $\mathbb{F}$ and an extension field $\mathbb{E}$. However, the choice of the central equation is far more restrictive than in HFE as we only have one monomial here, cf Section 3.1.3. We recall that the central polynomial has the form $P'(X') := X'^{q^\lambda+1}$ with the condition $\gcd(q^n - 1, q^\lambda + 1) = 1$. This condition is necessary to allow efficient inversion of $P'(X')$. Hence the overall "shape" of MIA is this degree $q^\lambda + 1$ and the fact that we only have a monomial rather than a polynomial. So in this setting, we cannot apply the additive sustainer, as this monomial does not allow any linear or constant terms. Moreover, the monomial requires a factor of one. Hence, we also have to preserve this property. The only sustainers suitable are the big sustainer (cf Sect. 5.2.2) and the Frobenius sustainer (cf Sect. 5.2.6). As we will see in Section 5.4, these are the only two sustainers applicable. Hence, we use both in the following

Theorem 5.3.7 *For $K := (S, P, T) \in Aff^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times Aff^{-1}(\mathbb{F}^n)$ a private key in MIA we have*

$$n(q^n - 1)$$

equivalent keys. Hence, the key-space of MIA can be reduced by this number.

Proof. To prove this statement, we consider normal forms of keys in MIA. In particular, we concentrate on a normal form of the affine transformation S where S is in univariate representation. As for HFE and w.l.o.g., let $B := S(1)$ be a non-zero coefficient on position 1. Unlike HFE we cannot enforce that $S(0) = 0$, so we may have $S(1) = 0$. However, in this case set $B := S(0)$. Applying $\sigma^{-1}(X) := B^{-1}X$ will ensure a normal form for S. In order to "repair" the monomial $P(X)$, we have to apply an inverse transformation to T. So let $\tau(X) := (B^{q^\lambda+1})^{-1}X$. This way we obtain

$$\begin{aligned}\mathcal{P} &= T \circ \tau^{-1} \circ \tau \circ P \circ \sigma \circ \sigma^{-1} \circ S\\ &= \tilde{T} \circ (B^{(q^\lambda+1).(-1)}.B^{q^\lambda+1}.X^{q^\lambda+1}) \circ \tilde{S}\\ &= \tilde{T} \circ P \circ \tilde{S}\,,\end{aligned}$$

where $\tilde{S}$ is in normal form. In contrast to HFE (cf Theorem. 5.3.1), we cannot chose the transformations σ and τ independently: each choice of σ implies a particular τ and vice versa. However, the fix point 1 is still preserved by the Frobenius sustainer and so we can apply this sustainer to the transformation S. As for HFE, we compute a normal form for a given generator and a total ordering of $\mathbb{E}$; again, we "repair" the monomial $X^{q^\lambda+1}$ by applying an inverse Frobenius sustainer to T and hence have

$$(BX^{q^c}, B^{-q^\lambda-1}X^{q^{n-c}}) \bullet (S, P, T) \text{ where } B \in \mathbb{E}^* \text{ and } 0 \leq c < n \text{ for } c \in \mathbb{N}\,,$$

which leads to a total of $n \cdot (q^n - 1)$ equivalent keys for any given private key. Since all these keys form equivalence classes of equal size, we reduced the private key space of MIA by this factor. □

Corollary 5.3.8 *For $K := (S, P, T) \in Aff^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times Aff^{-1}(\mathbb{F}^n)$ a private key in MIO (cf Section 6.4.1) we have*

$$n(q^n - 1)$$

equivalent keys. Hence, the key-space of MIO can be reduced by this number.

The above corollary can be proven in exactly the same way as Theorem 5.3.7. In particular, the fact that MIO is defined over odd rather than even characteristic does not impose a restriction in this context.

Remark 5.3.9 *Patarin observed that it is possible to derive equivalent keys by changing the monomial P [Pat96a]. As the aim of this chapter is the study of equivalent keys by chaining the affine transformations S,T alone, we did not make use of this property. A weaker version of the above theorem can be found in [WP05c]; in particular, it does not take the MIO class into account.*

Moreover, we observed in this section that it is not possible for MIA to change the transformations S,T from affine to linear. But from Section 4.5, we recall that Geiselmann et al. *showed how to reveal the constant parts of these transformations [GSB01]. Hence, having S,T affine instead of linear does not seem to enhance the overall security of MIA.*

For $q = 128$ and $n = 67$, we obtain $\approx 2^{469}$ equivalent private keys per class. The number of choices for S,T is $\approx 2^{63,784}$ in this case. This special choice of parameters has been used in Sflash^{v3} [CGP03a].

MIA-

As we recall from the cryptanalysis section, MIA itself is insecure, due to a very efficient attack by Patarin [Pat95]. However, for well-chosen parameters q, r, its variation MIA- (or C^{*--}) is believed to be secure: as in the case of HFE and HFE-, we use the original MIA scheme and apply the minus modification, cf sections 3.2.1 and 5.2.7.

Theorem 5.3.10 *For $K := (S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^n)$ a private key in MIA and a reduction number $r \in \mathbb{N}$ we have*

$$n.(q^n - 1)q^r \prod_{i=n-r-1}^{n-1} (q^n - q^i)$$

equivalent keys. Hence, the key-space of MIA- can be reduced by this number.

Proof. This proof is similar to the one of MIA, *i.e.*, we apply both the Frobenius and the big sustainer to S and the corresponding inverse sustainer to the transformation T. This way, we "repair" the change on the central monomial $X^{q^\lambda+1}$. All in all, we obtain a factor of $n \cdot (q^n - 1)$ equivalent keys for a given private key.

Next we observe that the reduction sustainer applied to the transformation T alone allows us to change the last r rows of the vector $v_T \in \mathbb{F}^n$ and also the last r rows of the matrix $M_T \in \mathbb{F}^{n \times n}$. This yields an additional factor of $q^r \prod_{i=n-r-1}^{n-1}(q^n - q^i)$ on this side.

Note that the changes on the side of the transformation S and the changes on the side of the transformation T are independent: the first computes a normal form for S while the second computes a normal form on T. Hence, we may multiply both factors to obtain the overall number of independent keys. □

For $q = 128, r = 11$ and $n = 67$, we obtain $\approx 2^{6180}$ equivalent private keys per class. The number of choices for S, T is $\approx 2^{63,784}$ in this case. This particular choice of parameters has been used in Sflash^{v3} [CGP03a].

5.3.3 Unbalanced Oil and Vinegar Schemes

We now move on to the Unbalanced Oil and Vinegar (UOV) class (see Section 3.1.1 for a formal definition). In contrast to the two schemes considered in the previous sections, UOV does not mix operations over two different fields $\mathbb{E}$ and $\mathbb{F}$ but only performs computations over the ground field $\mathbb{F}$. Hence, we have to consider different kind of sustainers. As we recall from previous sections, UOV omit the affine transformation T but use $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$. To fit in our framework, we set it to be the identity transformation, *i.e.*, we have $T = \tau = id$.

The "shape" of UOV is the fact that a system in the oil variables alone is linear. Hence, we may not mix vinegar variables into the set of oil variables. Still, there is an asymmetry here: while we may not "contaminate" the set of oil variables, there is no restriction on the set of vinegar variables: here, we may introduce *any* input variable. So for UOV, we can apply the additive sustainer and also the Gauss sustainer (cf sect. 5.2.1 and 5.2.5). However, in order to ensure that the shape of the central equations does not change, we have to ensure that the Gauss sustainer influences the oil variables independently from the vinegar variables.

Theorem 5.3.11 *Let $K := (S, P, id) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^n)$ be a private key in UOV. Then we have*

$$q^{n+mn} \prod_{i=0}^{n-m-1} (q^{n-m} - q^i) \prod_{i=0}^{m-1} (q^m - q^i)$$

equivalent keys. Hence, the key-space of UOV can be reduced by this number.

Proof. As in the case of the schemes before, we compute a normal form for a given private key. First, applying the additive sustainer reduces the affine transformation S to a linear transformation. This results in a factor of q^n in terms of equivalent keys. Second, applying the Gauss sustainer separately within vinegar and oil variables, we can enforce the following structure, denoted $R \in \mathbb{F}^{n \times n}$, on the matrix $M_S \in \mathbb{F}^{n \times n}$ of the (now only) linear transformation S:

$$R := \begin{pmatrix} I_m & 0 & A_m \\ 0 & I_{n-2m} & B_m^{n-2m} \\ 0 & 0 & I_m \end{pmatrix}.$$

In this context, the matrices I_m, I_{n-2m} are the identity elements of $\mathbb{F}^{m\times m}$ and $\mathbb{F}^{(n-2m)\times(n-2m)}$, respectively. Moreover, we have the matrices $A_m \in \mathbb{F}^{m\times m}$ and $B_m^{n-2m} \in \mathbb{F}^{(n-2m)\times m}$. For a given central equation $\mathcal{P}'$, each possible matrix R leads to the same number of equivalent keys. Let

$$E := \begin{pmatrix} F_{n-m} & 0 \\ G_{n-m}^m & H_m \end{pmatrix}$$

be an $(n \times n)$-matrix. Here, we require that the matrices $F_{n-m} \in \mathbb{F}^{(n-m)\times(n-m)}$ and $H_m \in \mathbb{F}^{m\times m}$ are invertible (cf Lemma 2.2.2). For $G_{n-m}^m \in \mathbb{F}^{m\times(n-m)}$, we have no restrictions. This way, we define the transformation $\sigma(x) := Ex$ where $x \in \mathbb{F}^n$. Note that these transformations σ form a subgroup within the affine transformations. So we have

$$(Ex + a, id) \bullet (S, \mathcal{P}', id) \text{ for } a \in \mathbb{F}^n \text{ and } E \text{ as defined above.}$$

As this choice of σ partitions the private key space into equivalence classes of equal size, and due to the restrictions on E, we reduced the size of the private key space by an additional factor of $q^{mn} \prod_{i=0}^{n-m-1}(q^{n-m} - q^i) \prod_{i=0}^{m-1}(q^m - q^i)$. $\square$

For $q = 2, m = 64, n = 192$, we obtain $2^{32,956}$ equivalent keys per key — in comparison to $2^{37,054}$ choices for S. If we increase the number of variables to $n = 256$, we obtain $2^{57,596}$ and $2^{65,790}$, respectively. Both choices of parameter have been used in [KPG03].

5.3.4 Stepwise-Triangular Systems

As pointed out previously, the UOV and the STS class are quite similar and both are from the single-field class. Therefore, the following proof on stepwise-triangular schemes uses the same ideas as the proof for the UOV class. As for UOV we exploit the fact that we can use Gauss operations within any given layer — and use again the fact that equations of layer l depend on all variables of the layers $1, \ldots, l$, *i.e.*, we may also perform Gauss operations on these previous layers, as long as the result only affects the given Layer l. We prove the following theorem for general STS, *i.e.*, stepwise triangular systems in their most general form, cf Section 3.1.2 for STS and its notation.

Theorem 5.3.12 *Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements, $n \in \mathbb{N}$ the number of variables, $m \in \mathbb{N}$ the number of equations and $L \in \mathbb{N}$ the number of layers. Moreover, let $(n_1, \ldots, n_L) \in \mathbb{N}^L$ be a vector of integers such that $n_1 + \ldots + n_L = n$ and $m_1, \ldots, m_L \in \mathbb{N}$ integers such that $m_1 + \ldots + m_L = m$. Then for $K :=$*

$(S, P, T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathbb{E}[X] \times \mathit{Aff}^{-1}(\mathbb{F}^n)$ *a private key in STS we have*

$$q^{m+n} \prod_{i=1}^{L} \left(q^{n_i(n - \sum_{j=1}^{i} n_i)} \prod_{j=0}^{n_i - 1} (q^{n_i - q^j}) \right) \prod_{i=1}^{L} \left(q^{m_i(m - \sum_{j=1}^{i} m_i)} \prod_{j=0}^{m_i - 1} (q^{m_i - q^j}) \right)$$

equivalent keys. Hence, the key-space of STS can be reduced by this number.

Proof. For this proof, we apply both the additive sustainer and the Gauss sustainer. The latter is applied independently on each layer.

First, we observe that we can apply the additive sustainer both to the transformation $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ to obtain their usual normal form $S(0) = T(0) = 0$. As a result, we obtain a factor of q^{m+n} and may assume $S \in \mathrm{Hom}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Hom}^{-1}(\mathbb{F}^m)$ for the remainder of this proof.

As in the proof of Theorem 5.3.11, we impose a special structure on the linear transformation S. Therefore, we consider the matrix

$$M_S := \begin{pmatrix} I_{n_1} & * & * & \cdots & & * & * \\ 0 & I_{n_2} & * & & & & * \\ 0 & 0 & I_{n_3} & & & & \\ & & & & & & \\ \vdots & & & \ddots & & & \vdots \\ & & & & & & \\ & & & & I_{n_{L-2}} & * & * \\ 0 & & & & 0 & I_{n_{L-1}} & * \\ 0 & 0 & & \cdots & 0 & 0 & I_{n_L} \end{pmatrix}$$

In $M_s \in \mathbb{F}^{n \times n}$, sub-matrices I_{n_i} are identity matrices in $\mathbb{F}^{n_i \times n_i}$ for $1 \leq i \leq n$. The left lower portion of M_S is zero while the upper right portion of M_S consists of elements of $\mathbb{F}$. To obtain this matrix M_S, we make use of

$$E := \begin{pmatrix} A_{n_1} & 0 & 0 & \cdots & & 0 & 0 \\ * & A_{n_2} & 0 & & & & 0 \\ * & * & A_{n_3} & & & & \\ & & & & & & \\ \vdots & & & \ddots & & & \vdots \\ & & & & & & \\ & & & & A_{n_{L-2}} & 0 & 0 \\ * & & & & * & A_{n_{L-1}} & 0 \\ * & * & & \cdots & * & * & A_{n_L} \end{pmatrix}$$

In this matrix $E \in \mathbb{F}^{n \times n}$, we have invertible components $A_{n_i} \in \mathbb{F}^{n_i \times n_i}$ for $1 \leq i \leq L$. Moreover, the upper right portion of the matrix E is zero while the left lower portion of E consists of elements of $\mathbb{F}$. We see that the above matrix is sufficient to impose this special structure on M_S. Moreover, for each choice of E, we obtain another linear transformation S and hence, M_S is a normal form of S.

Counting the number of possible matrices E we obtain a total of

$$\prod_{i=1}^{L} \left(q^{n_i(n-\sum_{j=1}^{i} n_i)} \prod_{j=0}^{n_i-1} (q^{n_i - q^j}) \right)$$

possibilities (cf Lemma 2.2.2 for the count of invertible matrices). To see the correctness of the above computation, we specialise it for n_1: we see that the term $\prod_{j=0}^{n_1-1}(q^{n_1-q^j})$ computes the number of choices for the matrix A_{n_1} while $q^{n_1(n-n_1)}$ computes the number of choices in the $(n_1 \times (n-n_1))$ column over $\mathbb{F}$ below the matrix A_{n_1}. By induction on n_i we obtain the above formula. In particular, as M_S is in normal form, there exists exactly one matrix E of the above form for any given $S \in \mathrm{Hom}^{-1}(\mathbb{F}^n)$. Hence, we have established the existence of an equivalence class of this size.

The corresponding proof for the transformation T is analogous. We just have to replace variables by equations here. In particular, we can still add equations layer-wise. So we have

$$(Ex + a, E'x + a') \bullet (S, \mathcal{P}', T) \text{ for } a \in \mathbb{F}^n, a' \in \mathbb{F}^m \text{ and } E, E' \text{ defined as above.}$$

As this choice of σ, τ partitions the private key space into equivalence classes of equal size, and due to the restrictions on E, E', we reduced the size of the private key space by the above number. □

Corollary 5.3.13 *For regular STS with step-width $r \in \mathbb{N}$, $L \in \mathbb{N}$ layers and $n := Lr$ variables, the above formula simplifies to*

$$q^{2n} \left(\prod_{l=1}^{L} q^{r(n-(l-1)r)} \prod_{i=0}^{r-1} (q^r - q^i)^L \right)^2 .$$

Choosing a regular STS scheme and $q = 2, r = 4, L = 25, n = 100$, we obtain $2^{11,315}$ equivalent keys for each given private key. For comparison: the number of choices for the two affine transformations S, T is $2^{20,096}$. Changing the number of layers to 20, and consequently having $r = 5$, we obtain a total of $2^{11,630}$ equivalent keys. This special choice of parameters has been suggested in [KS04c].

5.4 Tightness for MIA and MIO

All theorems in this chapter suffer from the same problem: we do not know if the size-reductions are "tight", *i.e.*, if the sustainers applied are the only ones possible. In this section we proof that for the MIA/MIO class (cf sections 3.1.3 and 6.4.1), the big sustainer and the Frobenius sustainer are actually the *only* possible way to achieve equivalent keys for MIA and MIO. We recall that both classes use a finite field $\mathbb{F}$ with $q := |\mathbb{F}|$ elements and an extension field $\mathbb{E}$ of dimension n over $\mathbb{F}$. Over $\mathbb{E}$, they use the monomial $Y' := X'^{q^\lambda+1}$ as central equation for $1 \leq \lambda < n$. While MIA needs q to be even, MIO is defined for q being odd. The proof for the MIA case is based on a so far unpublished observation by Prof. Dobbertin. Its extension to the MIO class is due to the author.

The starting point of the proof is the following equation which needs to hold for any two equivalent keys for the MIA / MIO class as we may only use affine transformations to transfer one private key to another one (cf Definition 5.1.1).

$$X^{q^\lambda+1} = T \circ X^{q^\lambda+1} \circ S\,,$$

which we can rewrite as

$$X^{q^\lambda+1} \circ S^{-1} = T \circ X^{q^\lambda+1}\,.$$

Exploiting that affine transformations form a group (cf Section 2.2.3), this also applies to their univariate representation (cf Definition 2.2.6). We express the above equation as

$$\left(\sum_{i=0}^{n-1} B_i X^{q^i} + A\right)^{q^\lambda+1} = \sum_{i=0}^{n-1} \tilde{B}_i \left(X^{q^\lambda+1}\right)^{q^i} + \tilde{A}\,,$$

with the coefficients $A, \tilde{A}, B_i, \tilde{B}_i \in \mathbb{E}$. Note that we have $(A+B)^p = A^p + B^p$ in a finite field of characteristic p and consequently $(A+B)^q = A^q + B^q$ for $q = p^k$ and some $k \in \mathbb{Z}^+$. We now use a matrix representation of the above equation, similar to the matrix used by Kipnis and Shamir in their cryptanalysis of HFE, cf Section 4.6.1. This yields

$$\begin{pmatrix}
A^{q^\lambda+1} & AB_0^{q^\lambda} X^{q^\lambda} & AB_1^{q^{\lambda+1}} X^{q^{\lambda+1}} & \ldots & AB_{n-1}^{q^{\lambda+n-1}} X^{q^{\lambda+n-1}} \\
B_0 A^{q^\lambda} X & B_0^{q^\lambda+1} X^{q^\lambda+1} & B_0 B_1^{q^\lambda} X^{q^{\lambda+1}+1} & & B_0 B_{n-1}^{q^\lambda} X^{q^{\lambda+n-1}+1} \\
B_1 A^{q^\lambda} X^q & B_1 B_0^{q^\lambda} X^{q^\lambda+q} & B_1^{q^\lambda+1} X^{q^{\lambda+1}+q} & \ldots & B_1 B_{n-1}^{q^\lambda} X^{q^{\lambda+n-1}+q} \\
\vdots & & \vdots & \ddots & \vdots \\
B_{n-1} A^{q^\lambda} X^{q^{n-1}} & B_{n-1} B_0^{q^\lambda} X^{q^\lambda+q^{n-1}} & B_{n-1} B_1^{q^\lambda} X^{q^{\lambda+1}+q^{n-1}} & \ldots & B_{n-1}^{q^\lambda+1} X^{q^{\lambda+n-1}+q^{n-1}}
\end{pmatrix}$$

$$= \begin{pmatrix} \tilde{A} & 0 & & \dots & 0 \\ 0 & \tilde{B}_0^{q^\lambda+1} X^{q^\lambda+1} & 0 & & 0 \\ & 0 & \tilde{B}_1^{q^{\lambda+1}+q} X^{q^{\lambda+1}+q} & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \tilde{B}_{n-1}^{q^\lambda+(n-1)+q^{n-1}} X^{q^{\lambda+n-1}+q^{n-1}} \end{pmatrix} \quad (*)$$

As we work in $\mathbb{E}$ which has q^n elements, we can reduce all powers larger than or equal to q^n by $q^n - 1$.

Lemma 5.4.1 *For $\mathbb{F}$ a finite field with $q > 2$ elements, the MIA and the MIO class can only use the big sustainer and the Frobenius sustainer to derive equivalent private keys.*

Proof. For this proof we show that the equations given by $(*)$ imply that $A = 0$ and all B_i except one are zero. Note that $B_0 = \ldots = B_{n-1} = 0$ implies that $S(X)$ is no bijection anymore but the transformation $S(X) = A$ for any input $X \in \mathbb{E}$ and fixed $A \in \mathbb{E}$. Hence, there must exist at least one non-zero coefficient B_i. W.l.o.g., we assume that B_0 is non-zero. Moreover, we assume an extension field of dimension $n \geq 2$. Note that this lemma is trivially true for $n = 1$.

For the proof, we make use of the fact that we can reduce all powers in $\mathbb{E}$ by $q^n - 1$. For powers of the form q^i this means that we can reduce the power i by n, *i.e.*, all computations are done in the ring $\mathbb{Z}/n\mathbb{Z}$ and we can hence assume $0 \leq a, b, c, d < n$ in the sequel. Moreover, we can distinguish the following three types of equations in $(*)$:

1. Equations of the form $AB^{q^\lambda+a} + B_b^{q^b} A^{q^\lambda} = 0$ for $a + \lambda \equiv b \pmod n$. We call them *equations of type A*. Note that they are related to terms with monomial of the form X^{q^b} for $0 \leq b < n$.

2. Equations of the form $B_a^{q^{\lambda+a}} B_b^{q^b} = 0$ with the condition $a + \lambda \equiv b \pmod n$ on the powers. We call them *equations of Hamming weight 1* and say that they are *self-dual*. Note that each row / column in the above matrix contains exactly one equation of Hamming weight 1 and that they correspond to terms with a monomial of the form X^{2q^b} for $0 \leq b > n$. As we have $q > 2$ there is no reduction of the power here.

3. Equations of the form $B_a^{q^{\lambda+a}} B_b^{q^b} + B_c^{q^{\lambda+c}} B_d^{q^d} = 0$ with the following conditions on their powers: first, we have $a \neq b, c \neq d$, as we otherwise would include equations from the diagonal. Obviously, we cannot make the assumption anymore that the right-hand side is equal to zero in this case. Second, we have $a + \lambda \not\equiv b \pmod n$ and $c + \lambda \not\equiv d \pmod n$ as we obtain equations of Hamming weight 1 otherwise. Third, we need $a + \lambda \equiv d \pmod n$ and $c + \lambda \equiv b \pmod n$ to ensure that the powers in the monomial

$X^{q^b+q^d}$ actually match. We call the pair (a, b) the *dual* of the pair (c, d). Note that this relation is reflexive, *i.e.*, (c, d) is the dual of (a, b). We call these *equations of type B*.

Note that equations of type A and equations of Hamming weight 1 do not mix as we have $q > 2$. Moreover, equations of Hamming weight 1 may not lie on the diagonal as we would have $\lambda + a \equiv a \pmod n$ in this case and hence $\lambda \equiv 0 \pmod n$, but this violates $0 < \lambda < n$. So far, we did not include any equation from the diagonal in our analysis. We come back to them later.

Inspecting the equation $B_0^{q^\lambda} B_\lambda^{q^\lambda} = 0$ of Hamming weight 1, we see that it implies $B_\lambda = 0$ as we have $B_0 \neq 0$ (see above). In addition, this implies $A = 0$ as we have $AB_0^{q^\lambda} + B_\lambda^{q^\lambda} A^{q^\lambda} = 0$ as an equation of type A. For $n = 2$, we are done. For $n \geq 3$, we can now use all equations of type B of the form $B_0^{q^\lambda} B_b^{q^b} + B_c^{q^{\lambda+c}} B_\lambda^{q^\lambda} = 0$. We notice that we need to meet the following conditions: $b \neq 0, \lambda$ and $c \neq 0, \lambda$ but $c + \lambda \equiv b \pmod n$. We see that we can construct pairs (b, c) meeting this conditions for all $b \in \mathbb{Z}/n\mathbb{Z}\backslash\{0, \lambda, 2\lambda\}$ with $0 < b < n$. Using the above equation we have established that all coefficients $B_b = 0$ as $B_0 \neq 0$ and $B_\lambda = 0$. Note that $\lambda \not\equiv 2\lambda \pmod n$ as we have $0 < \lambda < n$. Moreover, $2\lambda \not\equiv 0 \pmod n$ is not true either, which we see with the following argument: due to the size condition on λ, we know that we need to have $2\lambda = n$ to make the above equation hold. We use the condition $\gcd(q^n - 1 q^\lambda + 1) = 1$ for MIA and $\gcd(q^n - 1q^\lambda + 1) = 2$ for MIO to show that $2\lambda = n$ is impossible. Therefore we observe that $(q^{2\lambda} - 1) = (q^\lambda + 1)(q^\lambda - 1)$, *i.e.*, the gcd condition is violated for $n = 2\lambda$.

All left to show is that the coefficient $B_{2\lambda}$ is also equal to zero. To this end, we use the equation $B_{2\lambda}^{q^{3\lambda}} B_0^{q^0} + B_{-\lambda}^{q^0} B_{3\lambda}^{q^{3\lambda}} = 0$ of type B. In order to force the coefficient $B_{2\lambda}$ equal to zero, we need $B_{-\lambda} = 0$ or $B_{3\lambda} = 0$. Therefore, we use the equation $B_{-\lambda} q^0 B_0 q^0 = 0$ of type Hamming weight 1. As we have $B_0 \neq 0$, this implies $B_{-\lambda}$ and hence $B_{2\lambda} = 0$.

We have now established that all coefficients $A = B_1 = \ldots = B_{n-1} = 0$. Using the equations on the diagonal, these conditions also propagate through to the coefficients of the affine transformation T, *i.e.*, to $\tilde{A}, \tilde{B}_a$ for $0 < a < n$. Given that all coefficients but B_0 are zero, all equations which have terms of the form $B_a B_b$ for $a \neq 0, b \neq 0$ on the left hand side are now also zero, *i.e.*, they do not influence the equations of the form $B_i^{q^{\lambda+i}} B_i^{q^i} = \tilde{B}_j^{q^{\lambda+j}} \tilde{B}_j^{q^j}$ for some i, j with $0 \leq i, j < n$. We can not assume $i = j$ here as the matrix on the right hand side may have been rotated by a constant $r \in \mathbb{N}$ with $0 \leq r < n$. This is equivalent to the application of a Frobenius transformation. Still, we established that S, T may have only one non-zero coefficient in their univariate representation. Therefore, we know that the big sustainer and the Frobenius sustainer are the only two sustainers applicable to $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems of the MIA and the

MIO type. □

Unfortunately, the above proof is not valid in the case $q = 2$. The reason is that the equations of type A and Hamming weight 1 are mapped to one type of equation, namely $AB_a^{q^{\lambda+a}} + B_b^{q^b} A^{q^\lambda} + B_{a-1}^{q^{\lambda+a-1}} B_{b-1}^{q^{b-1}} = 0$ for $a + \lambda \equiv b \pmod{n}$. All other powers are also reduced (mod n). However, as soon as we assume $A = 0$, the above equation collapses to the original equation of Hamming weight 1, and the rest of the proof is again applicable. Alternatively, we could assume that any $B_i = 0$, and derive a similar proof starting with equations of type B. This leads to the following

Corollary 5.4.2 *For $q = 2$, the affine transformation S in univariate representation either has all coefficients $A, B_0, \ldots, B_{n-1}$ not equal to zero or exactly one coefficient B_i non-equal to zero and all other coefficients equal to zero. The same condition holds for the coefficients $\tilde{A}, \tilde{B}_0, \ldots, \tilde{B}_{n-1}$ of the transformation T.*

Still, we could not derive a contradiction with the assumption that all of the above values are non-equal to zero, so we have to leave the proof for the case $q = 2$ as an open problem. However, due to the very high number of equations of $O(n^2)$ compared to only $O(n)$ free variables, we conjecture that the above lemma also holds for $q = 2$ although we expect a far more technical proof in this case.

Table 5.1: Summary of the reduction results of this thesis

Scheme (*Section*)	Reduction
UOV (*5.3.3*)	$q^{n+mn} \prod_{i=0}^{n-m-1}(q^{n-m} - q^i) \prod_{i=0}^{m-1}(q^m - q^i)$
STS (*5.3.4*)	$q^{m+n} \prod_{i=1}^{L} \left(q^{n_i(n-\sum_{j=1}^{i} n_i)} \prod_{j=0}^{n_i-1}(q^{n_i}-q^j)\right) \prod_{i=1}^{L} \left(q^{m_i(n-\sum_{j=1}^{i} m_i)} \prod_{j=0}^{m_i-1}(q^{m_i}-q^j)\right)$
MIA (*5.3.2*)	$n(q^n - 1)$
MIA- (*5.3.2*)	$n(q^{n-r} - 1)q^r \prod_{i=n-r-1}^{n-1}(q^n - q^i)$
HFE (*5.3.1*)	$nq^{2n}(q^n - 1)^2$
HFE- (*5.3.1*)	$nq^n(q^n - 1)q^{n-r}(q^{n-r} - 1) \prod_{i=n-r-1}^{n-1}(q^n - q^i)$
HFEv (*5.3.1*)	$n'q^{n+n'+vm}(q^{n'} - 1)^2 \prod_{i=0}^{v-1}(q^v - q^i)$
HFEv- (*5.3.1*)	$n'q^{r+2n'vn'}(q^{n'} - 1)^2 \prod_{i=0}^{v-1}(q^v - q^i) \prod_{i=n'-r-1}^{n'-1}(q^{n'} - q^i)$

5.5 Discussion

In this chapter, we showed through the examples of Hidden Field Equations (HFE), Matsumoto-Imai Scheme A (MIA), Unbalanced Oil and Vinegar schemes (UOV), and Stepwise-Triangular Systems (STS) that $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems allow many equivalent private keys and hence have a lot of redundancy in their key spaces, cf Table 5.1 for an overview and Table 5.2 for numerical examples; the symbols used in Table 5.1 are explained in the corresponding sections.

Table 5.2: Numerical examples for the reduction results of this thesis

Scheme	Parameters	Choices for S, T (in $\log_2$)	Reduction (in $\log_2$)
UOV	$q = 2, m = 64, n = 192$	37,054	32,956
	$q = 2, m = 64, n = 256$	65,790	57,596
STS	$q = 2, r = 4, L = 25, n = 100$	20,096	11,315
	$q = 2, r = 5, L = 20, n = 100$	20,096	11,630
HFE	$q = 2, n = 80$	12,056	326
HFE-	$q = 2, r = 7, n = 107$	23,108	2129
HFEv	$q = 2, v = 7, n = 107$	21,652	1160
HFEv-	$q = 2, n = 107$	22,261	1258
MIA	$q = 128, n = 67$	63,784	469
MIA	$q = 128, n = 67$	63,784	6173

We see applications of our results in different contexts. First, they can be used for memory efficient implementations of the above schemes: using the normal forms outlined in this chapter, the memory requirements for the private key can be reduced without jeopardising the security of these schemes. Second, they apply to cryptanalysis as they allow to concentrate on special forms of the private key: an immediate consequence from Section 5.2.1 (additive sustainers) is that HFE does not gain any additional strength from the use of affine rather than linear transformations. Hence, this system should be simplified accordingly. Third, constructors of new schemes may want to keep these sustaining transformations in mind: there is no point in having a large private key space — if it can be reduced immediately by applying sustainers. Moreover, the results obtained in this chapter shine new light on cryptanalytic results, in particular key recovery attacks: as each private key is only a representative of a larger class of equivalent keys, each key recovery attack can only recover it up to these equivalences.

We want to stress that the sustainers from Section 5.2 are probably not the only ones possible. We therefore invite other researchers to look for even more powerful transformations. The only case where we know for certain that we found all sustainers possible, is the MIO/MIA class, cf Section 5.4 for the corresponding proof. In addition, there are other multivariate schemes which have not been discussed in this thesis. We are confident that they can be analysed using similar techniques as outlined in this thesis but have to leave the concrete proof as an open problem.

Chapter 6

Interesting Variants

While cryptography has many fascinating aspects in its theory, one of the main questions for most cryptographic ideas is *how* to apply them in practice. The same is true for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes: although they can be a fascinating subject in their own right, it is even more fascinating to see them applied. In this chapter we therefore outline some practical instances of $\mathcal{MQ}$-schemes, either used in practice or proposed for practical usage. In particular, we will see how the different trapdoors and modifiers "click" in place to derive practical signature schemes. In addition, we see why they cannot be used for encryption schemes so far.

Our own contribution in this section is a discussion of possible application domains of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes (Section 6.3), and the development of two secure tweaks of Quartz, namely Quartz-7m and Quartz-513d. Moreover, we were the first to put the schemes Rainbow, enhanced TTS, and Tractable Rational Map (cf Section 6.2) into the taxonomy developed in Chapter 3. Parts of this chapter have been previously published in [Wol02a, Wol02b, WP04, WP05a, WP05d].

6.1 NESSIE Contributions

Following historical development, we start with $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic signature systems in the context of the NESSIE project [NES]. NESSIE stands for New European Schemes for Signatures, Integrity, and Encryption and aimed at a better understanding of cryptography. In particular, it dealt with several cryptographic primitives, *e.g.*, symmetric ciphers and cryptographic hash functions; another area were public key signature algorithms. We see that both $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic contributions to NESSIE used schemes from the "mixed field class", *i.e.*, HFE and MIA. As a side-note, we want to mention that both were developed by the group around Jacques Patarin. We start with the submission(s) from the MIA class.

6.1.1 Flash / Sflash

Both Flash and Sflash have been submitted for use in restricted environments, *e.g.*, smart cards. Here, we describe both algorithms and their modification during the NESSIE process. When not stated otherwise, this section is based on [CGP00a, CGP00c, CGP02]. In the final evaluation of the NESSIE, Sflash$^{\mathrm{v2}}$ was selected as a signature algorithm for special application domains.

Table 6.1: Parameters for the *first* version of Flash and Sflash

Parameter	Sflash	Flash
$q = \|\mathbb{F}\|$	$128 = 2^7$	$256 = 2^8$
$n = \partial i(t)$	37	
r (equations removed)	11	
$P(x)$	$x^{128^{11}+1}$	$x^{256^{11}+1}$
Signature Length	259 bits	296 bits
Private Key Size	0.35 KByte	2.75 KByte
Public Key Size	2.2 KByte	18KByte

Both Flash and Sflash are MIA- signature schemes and hence use a bijection as private polynomial. In contrast to schemes of the HFE class, the central polynomial P is not private, but publicly known. The reason is the fact that both signature algorithms have so many public equations removed that this is not expected to be a threat for Flash's or Sflash's security. As we see in Table 6.1, there is a markable difference in the public key size of Flash and Sflash. In fact, this difference is much higher than expected: as Flash is based on the finite field $\mathbb{F} = \mathrm{GF}(2^8)$ and has a public key size of 18 KByte, Sflash is based on the finite field $\mathbb{F} = \mathrm{GF}(2^7)$. Therefore we would expect a public key size of 18KByte$\frac{7}{8} \approx$ 16KByte. The reason for this difference is due to the "subfield trick" from Section 3.2.3, *i.e.*, the restriction of the coefficients in both affine transformations S and T to a subfield of $\mathbb{F}$. In this case, coefficients for these two transformations come from $\widetilde{\mathbb{F}} = \mathrm{GF}(2)$ rather than $\mathbb{F} = \mathrm{GF}(2^7)$. And in fact, 18KByte$\frac{1}{8} \approx$ 2.2KByte, so the public key size (and accordingly the private key size) are within the expected range.

However, due to the attack from [GM02] (cf Section 4.5), the submission of

Table 6.2: Parameters for the second version of Sflash

Parameter	Sflash
$q = \lvert\mathbb{F}\rvert$	$128 = 2^7$
$n = \partial i(t)$	37
r (equations removed)	11
$P(x)$	$x^{128^{11}+1}$
Signature Length	259 bits
Private Key Size	2.45 KByte
Public Key Size	15.4 KByte

Sflash was changed. Moreover, Martinet suggested in [Mar01] to concentrate on either Flash and Sflash but not both algorithms as they are very similar. Due to the shorter signature length, the higher speed and also the shorter public key, she suggested to concentrate on Sflash. According to [PBO$^+$02], NESSIE followed this suggestion, so Flash was not considered in the second phase of the evaluation process of NESSIE.

In fact, in order to avoid the attack from [GM02], Sflash does no longer use a subfield $\widetilde{\mathbb{F}}$ of the finite field $\mathbb{F}$ but chooses the coefficients for the two affine transformations S and T from the whole field $\mathbb{F} = \mathrm{GF}(128)$. The overall parameters for Sflash, version 2, are summarised in Table 6.2.

Development after NESSIE

After Sflash$^{\mathrm{v}2}$ was approved by NESSIE, [CGP03b] announced Version 3 of Sflash. The reason was the cryptanalytic result of [Cou04]. In particular, the number of equations has been increased in the third version, cf Table 6.3. We want to stress that [Cou04] was later revoked and hence, we may assume that Sflash$^{\mathrm{v}2}$ is still secure. All parameters presented here have been taken from [CGP03a]. Shortly afterwards, Geiselmann et al. showed in [GSB01] how the constant parts of the two affine transformations S,T can be recovered for Sflash$^{\mathrm{v}3}$. As a reaction, [CGP03b] was published where these constant terms were declared to be "semi-public" parameters. As correctly pointed out there, an attacker cannot make use of them to discover the *full* transformations $S, T \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ as they are independently generated from the (still secret) matrices of S and T. However, we find this tweak rather unsatisfactory and suggest to skip the constant terms altogether: the security of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes lies in their quadratic parts, not in their linear (or even) constant terms. Hence, having constant terms does only increase the memory and also the computational complexity of an

Table 6.3: Parameters for the third version of Sflash

Parameter	Sflash
$q = \|\mathbb{F}\|$	$128 = 2^7$
$n = \partial i(t)$	67
r (equations removed)	11
$P(x)$	$x^{128^{33}+1}$
Signature Length	469 bits
Private Key Size	7.8 KByte
Public Key Size	112.3 KByte

implementation of Sflash. Still, the designers of Sflash decided to *include* linear and constant terms.

Discussion

As [Mar01] points out, Sflash (she was dealing with the first version of Sflash) has very well chosen parameters to avoid all known attacks against it. In fact, the change from Version 1 of Sflash to Version 2 of Sflash as outlined above, follows exactly this path. We observe a similar behaviour for Sflash Version 3. So from our current knowledge, Sflash (both versions 2 and 3) is secure against all known attacks. On the other hand, it is very new and the last attack dates back only three years. So for high security applications, Sflash is probably too new to be used at present. Still, for low and medium range security, Sflash is an interesting option, due to its suitability for restricted environments.

In this context, we have one final comment: due to the choice of parameters, in particular q^r , Sflash is not suitable for an encryption scheme. Moreover, due to the known attacks against the MIA and in particular the MIA- class, it is not possible to derive a secure encryption scheme.

6.1.2 Quartz

In Section 3.2, we looked at two important modifications of $\mathcal{MQ}$-schemes, namely the minus (Section 3.2.1) and the vinegar (Section 3.2.7) modification. In particular, they can be used in the context of HFE and are then called HFE- and HFEv, respectively. In this section, we see how they can be combined to obtain a practical signature scheme, namely Quartz. It was submitted to NESSIE [NES] but rejected. The purpose of this section is to describe why it failed.

Historical Note

The design goal of Quartz was not only to withstand all known attacks but also to have good chances to withstand future attacks as well. So the parameters in Quartz have been chosen rather conservatively, which results in a rather long signature time, namely 10 s on average on a Pentium II 500 MHz [CGP01]. As we know now (October 2005), the choice of parameters was not conservative enough. We discussed this point in more detail in Section 4. When not stated otherwise, this section is based on [CGP01] and describes the second, revised version of Quartz. The changes made from the first version (cf [CGP00b]) to the second version of Quartz are not due to security problems. Quite the contrary, they were made to speed up the whole algorithm without jeopardising its security. In addition, they allow a security proof for Quartz [Cou01]. We want to mention that this security proof has been disputed in [Gra05]. In the same paper, a new security proof is given. Unfortunately, the corresponding construction is not as efficient as the one given in [Cou01]. Still, it is the only construction of a "Patarin Chained Construction", for which a non-disputed security proof exists.

System Parameters

As we see in Table 6.4, the signature length (128 bits) is 21 bits larger than expected: as the extension field $\mathbb{E}$ has dimension 103 and there are 4 vinegar variables added, we would expect a signature length of 107 bits. The reason for this difference lies in the fact that Quartz uses a so-called "Chained Patarin Construction" (CPC, called "Feistel-Patarin network" in the submission to NESSIE) to compute the signature. Within this construction, the HFE algorithm is called four times to compute a signature, *i.e.*, this involves solving the underlying HFE problem four times (cf Table 6.3 and Section 6.1.2). This way, we need to add 4 times 7 bits to the number of public equations and obtain a signature length of $100 + 4 \cdot 7 = 128$ bits.

System Description

To deal with the security features of Quartz, we try to deduce if they enhance or jeopardise the security of Quartz. First of all, the private polynomial P has full coefficients, *i.e.*, it has non-trivial coefficients from $\mathbb{E}$ and also all possible coefficients, *i.e.*, every power which has Hamming weight two or lower up to degree 129. Together with the vinegar variables (denoted $z_1, \ldots, z_4$), the private

Table 6.4: Parameter for Quartz

Parameter	Quartz
$q = \|\mathbb{F}\|$	2
$n = \partial i(t)$	103
transformation S	$\mathbb{F}^{107} \to \mathbb{F}^{107}$
transformation T	$\mathbb{F}^{103} \to \mathbb{F}^{103}$
l (equations removed)	3
v (vinegar variables)	4
m (equations)	100
n (variables)	107
d (degree)	129
Signature Length	128 bits
Private Key Size	3 KByte
Public Key Size	71 KByte

polynomial P of Quartz can be expressed as:

$$\begin{aligned} P_{(z_1,\ldots,z_4)}(x) \quad := \quad & \sum_{\substack{0 \le i,j \le 7 \\ q^i+q^j \le 129}} Ci,jx^{q^i+q^j} + \sum_{\substack{0 \le k \le 7 \\ q^k \le 129}} B_k(z_1,\ldots,z_v)x^{q^k} \\ & +A(z_1,\ldots,z_v) \\ & \text{for } C_{i,j} \in \mathbb{E}, \\ & B_k(z_1,\ldots,z_v) \text{ are affine in } (z_1,\ldots,z_v)\text{, and} \\ & A(z_1,\ldots,z_v) \text{ is at most quadratic in } (z_1,\ldots,z_v) \end{aligned}$$

As the polynomial has all coefficients and the degree is rather high, Quartz withstood at design time all known attacks up to a complexity level of 2^{80} 3-DES computations — this level was requested for signature algorithms in NESSIE. This is also true if there is no v modification. In fact, the degree is very high as in 2000 a degree of 25–33 was believed to be sufficient. In addition, Quartz is a HFEv rather than a "basic" HFE scheme. This modification was expected to further enhance the security of Quartz. Moreover, Quartz is also a HFE- scheme with three equations kept secret. As Quartz uses a very general polynomial P and also the v modification, the attacks known against basic HFE do not apply against Quartz. So removing only three equations from the public key seemed sufficient for Quartz and actually enhanced its overall security against all attacks known so far. We call the parts of Quartz discussed above the "HFE"-step, *i.e.*,

$\mathrm{HFE}(x) := T \circ P \circ S(x)$ and its inverse $\mathrm{HFE}^{-1}(y) := S^{-1} \circ P^{-1} \circ T^{-1}(y)$.

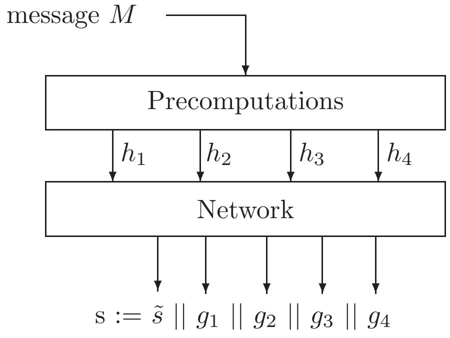

Figure 6.1: Overall Structure of Quartz for Signature Generation

Precomputations and HFE-step

Although the HFE-step itself looks quite secure, there is an obvious attack using the birthday paradox: by computing 2^{50} different versions of the message M and by applying the public key to 2^{50} different values for $x_1, \ldots, x_n \in \mathrm{GF}(2)$ we expect to obtain a valid signature for one version of the message M. This is true as the HFE step alone uses only 100 public equations over GF(2) in 100 variables each. Hence, we can expect a collision after $2^{100/2} = 2^{50}$ steps. The general principle is called a "birthday attack" [MvOV96, Sect. 9.7.1]. This is far less than the complexity level of 2^{80} required in NESSIE. To overcome this problem, Quartz combines four invocations of the HFE-step in a so-called "Chained Patarin Construction". The key idea of this network is not to store four times a full signature (*i.e.*, a signature of 428 bits in total) but to save only the last signature completely. In addition, it stores 7 bits for each of the 4 signatures computed. The reason for this lies in the fact that the HFE-step of Quartz has only 100 bits of input but a 107 bit output. These additional 7 bits compensate for this expansion. The overall structure of Quartz is shown in Figure 6.1. As we see there, signature generation with Quartz requires a precomputation step (see Figure 6.2) before applying the Chained Patarin Construction itself (see Figure 6.3). The key idea of the precomputation step is to use three calls of a 160-bit hash function (SHA-1 in Quartz, cf [FIP] for SHA-1) to "expand" a 160-bit hash (denoted m_0 in Figure 6.2) to four 100-bit values h_1, h_2, h_3 and h_4. During this process, the original hash m_0 is concatenated (operator $\cdot||\cdot$) with the 8 bit values $0x00$, $0x01$ and $0x02$ (notation for the hexadecimal numbers 0, 1 and 2, *e.g.*, in C, C++, or Java) to obtain three 168 bit values. Each of them is hashed individually using a 160 bit hash function and then concatenated. The

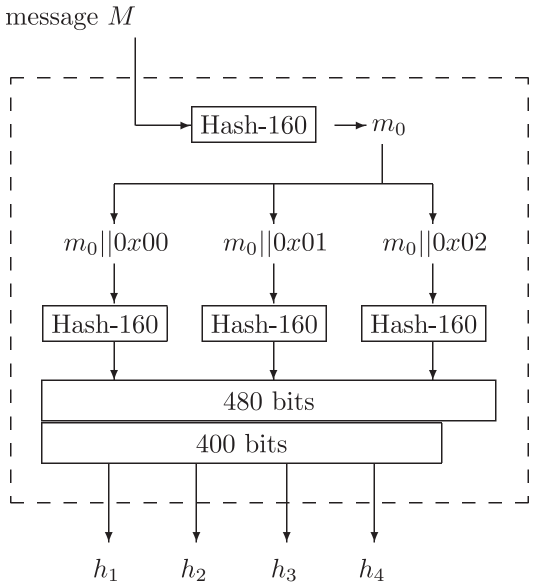

Figure 6.2: Precomputation in Quartz

resulting 480 bit number is truncated to 400 bits and yields four 100-bit strings. If Quartz used a hash function with a 512 bit output rather than 160 bit, the precomputation step would be obsolete. Such functions have been accepted in the NESSIE project (*e.g.*, algorithm "Whirlpool") and are also suggested by the NIST (only the SHA-512 algorithm in [FIP01]). But for the complexity level of 2^{80}, it is sufficient to use a 160 bit hash function and to expand its output to 400 bits as done in the precomputation step of Quartz. Still, in the light of recent attacks against hash functions, it seems highly advisable to move from SHA-1 to SHA-512. On the other hand, these attacks were not known during the NESSIE evaluation and hence, the choice of SHA-1 was adequate at design time.

Chained Patarin Construction

We now concentrate on the "Chained Patarin Construction" itself as outlined in Figure 6.3. It uses the output of the precomputation step as input. We first describe the initial step of the network. After loading the counter i with the value 0, it "xors" h_0 with 0 to obtain the intermediate value y. This is certainly obsolete as h_0 "xor" $0 = h_0$. However, during the run of the algorithm, h_1, h_2, h_3 are "xored" with the output of the previous step, so this "xor" operation is required for symmetry of the four steps. After this initialisation, the 100-bit

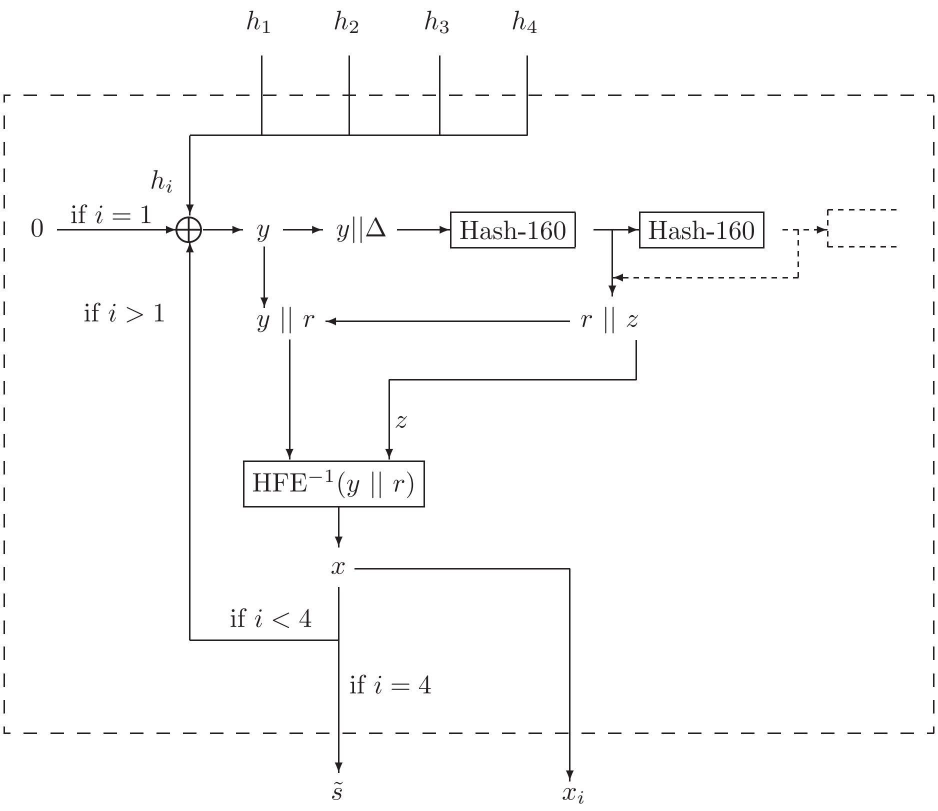

Figure 6.3: Central Structure of the Chained Patarin Construction for Quartz

value y is hashed together with a secret 80 bit parameter Δ to obtain the random variables r (3 bit) and the vinegar variables z (4 bit). Both are fed into the HFE step to obtain a valid signature of 107 bits. According to [CGP01, Sec. 5.3] the probability to obtain a valid signature at the first attempt is $\approx 60\%$. This idea has already been introduced in Section 2.3.2 of this thesis. If there is no valid signature, the hash of $(y \mid\mid \Delta)$ is rehashed. This is repeated until a valid signature is obtained. The probability that there is no valid signature at all for a given message M is estimated to be $\leq 2^{-183}$ and hence negligible [CGP01, Sec. 5.3]. If a valid 107 bit signature is found, the least significant 100 bits of x are fed back into the network while the most significant 7 bits are stored as output g_1. The other three steps are similar but h_i is not "xored" with 0 but with the least significant 100 bits of x. In the final step, these 100 bits are not fed into the network but yield the output $\tilde{s}$. In each step, it is possible that there is not only one, but up to $d = 129$ different solutions for the equation $x = HFE(y \mid\mid r)$. The Quartz-specification states that only one is chosen, namely the one with the least hash value (bit-wise comparison without sign bit, most-significant-bit being on the left-hand side).

Signature Verification

In order to verify the validity of a signature, this network is reversed. As the public key consists of 100 polynomials $p_1, \ldots, p_{100}$ in 107 input variables $x_1, \ldots, x_{107}$, the 7 bit values $g_1, \ldots, g_4$ are used to obtain 107 bits input for the public key during each run. In addition, as the four 100 bit values $h_1, \ldots, h_4$ are "xored" each time, a signature is only valid if the overall output of this scheme is 0. In this case the signature is accepted.

Note that during signature verification, it is not possible to check if *the* solution with the smallest hash value has been chosen. Hence, for an attack against Quartz, it is sufficient to compute *any* solution $x \in \mathbb{F}^n$ for the equation $y =$ HFE(x). Due to the Chained Patarin Construction, we need to compute such a solution four times.

Discussion

The Chained Patarin Construction is certainly a rather complicated security feature. However, as each signature depends on a 400 bit input (which is obtained from a 160 bit hash value), it seemed to be a rather strong signature system. Moreover, as Quartz uses the 160-bit hash function as a kind of cryptographically secure random number generator, it is deterministic, so each message has always the same signature (for the same private key K). In fact, there is no known attack against the CPC so far. In the original specification of HFE as a signature scheme it seemed to be necessary to use "real" randomness to obtain valid signatures. As real randomness often is a problem (*e.g.*, in a stand-alone server without user interaction), the deterministic version using a pre-stored 80-bit secret, makes it possible to use Quartz in more application domains. On the other hand, we saw that the security proof of Quartz is unfortunately flawed [Gra05]. Therefore it seems advisable to replace the Chained Patarin Construction with the new construction from [Gra05] — or to fix the flaw in the security proof.

As we saw in the Section 4.6, Quartz as proposed in [CGP01] can no longer considered to be secure; main reason is the attack from [FJ03]. Consequently, it has not been recommended by NESSIE [PBO$^+$03]. In fact, this attack on its HFE step is the reason that Quartz failed in the end.

Due to its choice of parameters, we would need $2^7 \cdot 10$ seconds, *i.e.*, around 20 minutes if we were to use Quartz in an encryption scheme. This obviously too long. The value of 10 seconds on a Pentium II with 500 Mhz for one inversion of Quartz has been reported in [CGP01].

Secure Versions of Quartz

Assuming that there is a fix for the security proof from [Cou01], we see at present two possibilities to obtain a secure version of the Quartz signature algorithm which is able to withstand the attack from [FJ03]. The first uses a degree of 513 for the public polynomial and keeps the other parameters unchanged. We call this version Quartz-513d and expect an attack complexity of $\approx 2^{82}$. However, due to the very large degree of the private polynomial, and hence the also high signature generation time, we do not expect this version to be of practical interest.

Therefore, we concentrate on a different modification: replace the 4 vinegar variables by removing 4 equations. The corresponding system has still the same signature size as Quartz but an estimated attack complexity of $\approx 2^{86}$. It therefore meets the NESSIE-requirements of 2^{80} 3-DES-computations.

Table 6.5: Parameter for different versions of Quartz

Parameter	Quartz	Quartz-7m	Quartz-513d
$q = \|\mathbb{F}\|$	2		
$\partial i(t)$ (degree $\mathbb{E}$)	103	107	103
transformation S	$\mathbb{F}^{107} \to \mathbb{F}^{107}$		
transformation T	$\mathbb{F}^{103} \to \mathbb{F}^{103}$	$\mathbb{F}^{107} \to \mathbb{F}^{107}$	$\mathbb{F}^{103} \to \mathbb{F}^{103}$
l (equations removed)	3	7	3
v (vinegar variables)	4	0	4
m (equations)	100		
n (variables)	107		
d (degree)	129	129	513
Signature Length	128 bits		
Private Key Size	3 KByte	3KByte	4KByte
Public Key Size	71 KByte		
Security Level	2^{62}	2^{82}	2^{86}

Although these versions (cf Table 6.5) are secure against the recent attack from Faugère and Joux, we argue to be cautious as they have not been independently studied by other researchers. It is therefore well possible that they carry unnoticed weaknesses.

6.2 Mixed Schemes

After concentrating on two examples from the mixed *field* class, we now move on to examples of the mixed *schemes* class. In total, we describe three interesting, but rather new schemes, namely *enhanced TTS* [YC04b], *tractable rational map* [WHL+05], and *Rainbow* [DS05b]. All three schemes use an STS structure as overall layout and "plug in" trapdoors of other schemes in the individual layers. Hence, they do not use STS in its "pure" form but mix it with other trapdoors. This motivates the name "mixed schemes". As we saw in Section 4.4, pure STS can neither be used for signature nor encryption schemes. So we consider this new way of using STS a clever move to cut down memory and computational complexity when implementing $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes in practice.

6.2.1 Enhanced TTS

In [YC04b], Yang and Chen give several constructions of the so-called *enhanced TTS* schemes (enTTS). For all these schemes, only the central equations $\mathcal{P}'$ change. We concentrate on their first proposal as all other schemes developed only vary the security parameters, but keep the same idea, *i.e.*, using an overall STS structure with an UOV trapdoor in each layer. For this construction, they use the following central polynomials, cf Figure 6.4. Here we have $\gamma'_{i,j} \in_R \mathbb{F}$ random

$$\begin{array}{rcl}
p'_i & := & x'_i + \sum_{j=1}^{7} \gamma'_{i,j} x'_j x'_{8+(i+j \bmod 9)} \text{ , for } i = 8 \ldots 16;\\
p'_{17} & := & x'_{17} + \gamma'_{17,1}x'_1x'_6 + \gamma'_{17,2}x'_2x'_5 + \gamma'_{17,3}x'_3x'_4 + \gamma'_{17,4}x'_9x'_{16} +\\
 & & +\gamma'_{17,5}x'_{10}x'_{15} + \gamma'_{17,6}x'_{11}x'_{14} + \gamma'_{17,7}x'_{12}x'_{13};\\
p'_{18} & := & x'_{18} + \gamma'_{18,1}x'_2x'_7 + \gamma'_{18,2}x'_3x'_6 + \gamma'_{18,3}x'_4x'_5 + \gamma'_{18,4}x'_{10}x'_{17} +\\
 & & +\gamma'_{18,5}x'_{11}x'_{16} + \gamma'_{18,6}x'_{12}x'_{15} + +\gamma'_{18,7}x'_{13}x'_{14};\\
p'_i & := & x'_i + \gamma'_{i,0}x'_{i-11}x'_{i-9} + \sum_{j=19}^{i} \gamma'_{i,j-18}x'_{2(i-j)-(i \bmod 2)}x'_j +\\
 & & +\gamma'_{i,i-18}x_0, x_i + \sum_{j=i+1}^{27} \gamma'_{i,j-18}x'_{i-j+19}x'_j \text{ , for } i = 19 \ldots 27.
\end{array}$$

Figure 6.4: Central Map for enhanced TTS

coefficients. We note that the central polynomials do not have linear or constant random terms. As the security of the $\mathcal{MQ}$-problem lies in the quadratic part of these equations alone, this is certainly a good idea as it saves both evaluation time and private key memory.

Having a closer look at the polynomials $p'_8, \ldots, p'_{16}$ we see that they only depend on the input variables $x'_1, \ldots, x'_{16}$, and hence they form the first layer of an STS scheme. The second layer is formed by the two polynomials p'_{17}, p'_{18}, which also depend on x'_{17}, x'_{18}, and the last layer is formed by $p'_{19}, \ldots, p'_{27}$, which depend on *all* 28 input variables $x'_0, \ldots, x'_{27}$.

For inverting this trapdoor function, we first assign random values to $x'_1, \ldots, x'_7$, which gives a degree 1 system of equations in $y'_i = p'_i$ for $i = 8 \ldots 16$. Note that we do not always get a solution here. However, in this case we just assign new random values to $x'_1, \ldots, x'_7$ and try again (cf [YC04b] for more details). Second, we notice that the variables x'_{17} and x'_{18} are free variables in the polynomials p'_{17} and p'_{18}. Hence, no matter which value the other terms in these two polynomial have, we can always choose x'_{17} and x'_{18} such that the corresponding equations are satisfied. Hence, there will always be a solution at this stage. The final step is to assign a random value to the variable x'_0, which guarantees a solution at this level of the internal equations, too. Hence, we see that the overall structure of enTTS follows UOV. However, in order to speed up computations and to save memory, the equations have been made very sparse (see Figure 6.4). We want to point out that this sparsity gave some unexpected structure and hence allowed the author of [DY04] to break an earlier version of the scheme which was presented in [YC04a]. The only difference between [YC04a] and [YC04b] is a slight modification in the last block of equations, see first summation and the missing term between the two summations:

$$\begin{aligned} p'_i \quad := \quad & x'_i + \gamma'_{i,0} x'_{i-11} x'_{i-9} + \sum_{j=19}^{i} \gamma'_{i,j-18} x'_{2(i-j)} x'_j + \\ & + \sum_{j=i+1}^{27} \gamma'_{i,j-18} x'_{i-j+19} x'_j \text{ , for } i = 19 \ldots 27. \end{aligned}$$

We see that this is only a very small modification. Still, it is able to overcome the attack from [DY04]. At present, we are not aware of any weaknesses of enTTS.

Taking a second look at the scheme from Figure 6.4, we see that the two polynomials p'_{17} and p'_{18} actually can further be classified as UOV with two branches: while p'_{17} does not depend on x'_{18}, the formula for p'_{18} is independent from x'_{17}. This is the reason that we can say enTTS uses a kind of branching structure for these two polynomials. However, as the overall scheme uses more an STS structure, this small branching part cannot be used to launch the attacks mentioned in Section 3.2.4.

As for the schemes discussed so far, we cannot use enTTS for encryption, as inverting the UOV step requires too many computations, namely $2^{8\cdot 11} = 2^{88}$. This is clearly above our computational threshold of 2^{80} 3-DES computations.

6.2.2 Tractable Signature Schemes

After enTTS, we move on to the tractable signature scheme from [WHL$^+$05], which is again a scheme with an STS structure. This time, it uses a total of five layers. However, the twist in comparison with a normal STS scheme lies in the fact that computations in the different layers are done in extension fields $\mathbb{E}_l$ for $l = 2 \dots 5$ rather than in the ground field $\mathbb{F}$ only. In the following, we denote with "$\cdot$" multiplication in the corresponding extension field and with ϕ_l for $l = 2 \dots 5$ the corresponding canonical bijection (cf Definition 2.1.6).

First Layer. The first layer uses the variables $x'_1, \dots, x'_8$ as an input and polynomials of the form $p'_i := x'_i$, *i.e.*, we have the simplest polynomials possible for our purpose. Moreover, we assign random values to these variables and hence, there is no need for an extension field in this layer.

Second Layer. We have $\mathbb{E}_2 = \mathbb{F}^6$ and the second layer as

$$\mathcal{P}'_2 := \phi(\phi^{-1}(x'_9, \dots, x'_{14}) \cdot \phi^{-1}(x'_1, \dots, x'_6)) + \begin{pmatrix} c'_1 x'_1 x'_2 \\ c'_2 x'_2 x'_3 \\ \vdots \\ c'_6 x'_6 x'_7 \end{pmatrix} + \begin{pmatrix} c'_7 x'_3 \\ c'_8 x'_4 \\ \vdots \\ c'_{12} x'_8 \end{pmatrix}.$$

We notice that the second layer becomes linear if the variables $x'_1, \dots, x'_6$ are given. In addition, we have $c'_1, \dots, c'_{12} \in_R \mathbb{F}$ random coefficients.

Third Layer. We have $\mathbb{E}_3 = \mathbb{F}^2$ in the third layer

$$\begin{aligned} \mathcal{P}'_3 \quad := \quad & \phi([\phi^{-1}(x'_{15}, x'_{16})]^2) \\ & + \begin{pmatrix} c'_{13} x'_1 x'_2 + c'_{14} x'_3 x'_4 + \dots + c'_{19} x'_{13} x'_{14} \\ c'_{20} x'_{14} x'_1 + c'_{21} x'_2 x'_3 + \dots + c'_{26} x'_{12} x'_{13} \end{pmatrix} + \begin{pmatrix} c_{27} x'_1 \\ c_{28} x'_2 \end{pmatrix}. \end{aligned}$$

At first glance, the new variables x'_{15}, x'_{16} do not introduce a permutation. For brevity, we write $X' := \phi^{-1}(x'_{15}, x'_{16})$. However, as the above construction is only specified over fields of characteristic 2, we have X'^2 being a bijection. Unfortunately, [WHL$^+$05] does not go into details how to invert this function, but assuming $\gcd(2, q^2 - 1)) = 1$ as for the parameters proposed in [WHL$^+$05], we can use the same technique as for the MIA trapdoor (cf Section 3.1.3) to invert the function $Y' = X'^2$ for given $Y' \in \mathbb{E}_3$ and unknown $X' \in \mathbb{E}_3$.

Again we notice that this bijection does not depend on the variables of the previous layers, *i.e.*, on $x'_1, \dots, x'_{14}$. Moreover, we have $c'_{13}, \dots, c'_{28} \in_R \mathbb{F}$ random coefficients.

Fourth Layer. We have $\mathbb{E}_4 = \mathbb{F}^3$ here and

$$\begin{aligned}\mathcal{P}'_4 &:= \phi(\phi^{-1}(x'_{17}, x'_{18}, x'_{19}) \cdot \phi^{-1}(x'_8, x'_9 + x'_{11} + x'_{12}, x'_{13} + x'_{15} + x'_{16})) \\ &\quad + \begin{pmatrix} c'_{29}x'_4x'_{16} \\ c'_{30}x'_5x'_{10} \\ c'_{31}x'_{15}x'_{16} \end{pmatrix} + \begin{pmatrix} c'_{32}x'_9 \\ c'_{33}x'_{10} \\ c'_{34}x'_{11} \end{pmatrix}.\end{aligned}$$

We have $c'_{19}, \dots, c'_{34} \in_R \mathbb{F}$ random coefficients. Moreover, we notice that the fourth layer becomes linear if the old variables $x'_1, \dots, x'_{16}$ are given.

Fifth Layer. We have $\mathbb{E}_5 = \mathbb{F}^9$ and

$$\begin{aligned}\mathcal{P}'_5 &:= \phi(\phi^{-1}(x'_{20}, x'_{21}, \dots, x'_{28}) \\ &\quad \cdot\phi^{-1}(x'_1, x'_2 + x'_6 + x'_{11}, x'_3 + x'_7 + x'_{12}, x'_4 + x'_8 + x'_{13}, x'_5 + x'_9 + x'_{14}, \\ &\quad x'_{10} + x'_{14} + x'_{16}, x'_{11} + x'_{15} + x'_{17}, x'_{12} + x'_{16} + x'_{18}, x'_{13} + x'_{17} + x'_{19})) \\ &\quad + \begin{pmatrix} c'_{35}x'_{18}x'_{19} \\ c'_{36}x'_{17}x'_{13} \\ c'_{37}x'_{16}x'_{14} \\ c'_{38}x'_{12}x'_{13} \\ c'_{39}x'_{15}x'_{14} \\ c'_{40}x'_{19}x'_{12} \\ c'_{41}x'_{18}x'_{10} \\ c'_{42}x'_{12}x'_6 \\ c'_{43}x'_{13}x'_5 \end{pmatrix} + \begin{pmatrix} c'_{44}x'_1 \\ c'_{45}x'_2 \\ \vdots \\ c'_{52}x'_9 \end{pmatrix}.\end{aligned}$$

We have $c'_{35}, \dots, c'_{52} \in_R \mathbb{F}$ random coefficients. Moreover, we notice that this last layer becomes linear if the old variables $x'_1, \dots, x'_{19}$ are given.

As an overall result, we see that the tractable rational map signature scheme is an instance of an STS scheme with sparse polynomials. In contrast to the enhanced TTS from the previous sections, these polynomials are over different extension fields rather than the ground field. Hence, these extension fields have to be chosen carefully to allow fast multiplication and inversion. We refer to [WHL$^+$05] for details on these choices. Using the taxonomy developed in this thesis, we see that the first and the second layer can actually be combined to one: we view this new layer one/two as a UOV step with $x'_1, \dots, x'_7$ the vinegar and $x'_9, \dots, x'_{14}$ the oil variables.

[WHL$^+$05] claims that all known attacks have been taken into account for this construction and it does not cover any hidden weakness. As for enhanced

TTS, we suggest to wait a while until using this construction as the sparsity of the polynomials may open the door for previously unknown attacks, in particular as the corresponding encryption scheme from [WC04] has been successfully cryptanalysed in [JKJMR05], using observations on the linearity of the overall system. Using the proofs from [JKJMR05], we expect Gröbner attacks to have a rather low running time, too, against the scheme from [WC04]. However, the attacks from [JKJMR05] do not extend to [WHL$^+$05].

The version [WHL$^+$05] has the unfortunate property that we may not obtain a valid signature with the first choice of random variables in all cases; we already noticed a similar behaviour for enhanced TTS, see above. To verify that tractable rational maps have this problem, too, we observe that $x'_1 = \ldots = x'_8 = 0$ is a valid assignment in the first layer. Now, in the second layer, the multiplication in the extension field $\mathbb{E}_2$ always yields 0. Hence, no matter which values we choose for $x'_9, \ldots, x'_{14}$, we cannot fulfil the equations $y'_9 = p'_9, \ldots, y'_{14} = p'_{14}$ for $y'_9, \ldots, y'_{14}$ all non-zero. Although the probability for such a behaviour is rather low (2^{-64}), it is not zero and hence, tractable rational maps do not have a constant signing time. We can draw similar conclusions for Layer 4: assume we have $x'_8 = x_9 + x'_{11} + x'_{12} = x'_{13} + x'_{15} + x'_{16} = 0$. This event happens with probability 2^{-24}. Moreover, assume that the other terms in the corresponding equation are *non-zero*. This happens with probability $1 - 2^{-24}$. Now, we cannot compute a valid solution in the fourth layer and hence, cannot find a valid signature. Note that we do not encounter such a problem in the third layer as we can always compute the inverse of $[\phi^{-1}(x'_{15}, x'_{16})]^2$ for any given input. In any case: both problems were independently noticed by the author of [WHL$^+$05] who suggested the following tweak in their presentation at PKC 2005 [Wan05]: instead of assigning random values to all variables, they suggest to select the first 8 variables $x'_1, \ldots, x'_8 \in_R \mathbb{F}^*$, *i.e.*, without the possibility of getting 0 for any of these hidden variables. This way, they can guarantee that for each try they obtain a valid signature. While we do not expect security problems here, we find it too drastic a solution: just enforcing $x'_1 \neq 0$ and $x'_8 \neq 0$ would have been sufficient to overcome the problem of not obtaining a valid signature in all tries.

In any case: both mixed schemes are rather complicated to cryptanalyse as they use very specific polynomial equations. In particular for the latter scheme, the rationals behind choosing specific structures has not been made explicit. Hence, it is difficult for an outsider to judge if these choices are in fact rational or not. In particular, some more explanation by the author of [WHL$^+$05] would certainly help here.

As for the schemes discussed so far, we cannot use tractable rational maps for encryption, as inverting the first step requires too many computations, namely $2^{8 \cdot 11} = 2^{88}$. This is clearly above our computational threshold of 2^{80} 3-DES computations. Moreover, the encryption scheme from [WC04] has been successfully broken in [JKJMR05], which further indicates a security problem of encryption schemes of the tractable map class.

6.2.3 Rainbow

This scheme has been suggested in [DS05b]. Following our classification, it is an STS construction which uses an UOV trapdoor at each layer. In the first layer, the vinegar variables are assigned randomly. In all next layers, the vinegar variables are the variables from the previous layers. As for the two schemes discussed above, it may happen that we do not obtain a solution for a given set of vinegar and oil variables. As above, we simply try again in this case.

From a security point of view, the ideas of rainbow are sound. In particular the suggested sets of parameters in [DS05b, Sec. 2.2] take all practical attacks into account. Here we have a ground field of size $q = 256$, a total of $n = 33$ variables in $m = 27$ equations, and four layers. The first layer has six oil and six vinegar variables, the second and the third layers add both five new (oil) variables, and the final fourth layer adds additionally 11 oil variables. The latter is due to rank attacks as discussed, *e.g.*, in [WBP04].

There is also an improved version of Rainbow, discussed in [DS05b, Sec. 5]. Again we have $q = 256$, $n = 33$ and $m = 27$. Moreover, the number of variables in the first and the last layer are the same. However, all intermediate layers now only have one new variable each. Hence, the number of layers increases accordingly. This way, we always obtain a solution at these intermediate layers. Therefore, the overall number of repetitions until we get a valid signature drops considerably. A more detailed discussion of this idea can also be found in Section 6.4.2.

As Rainbow is derived from the UOV class, it is clearly not feasible to derive a secure encryption scheme with this construction.

6.2.4 Discussion

We notice that all three schemes use an overall STS structure to obtain a secure construction for signature generation. In addition we see that they guide their initial layer, *i.e.*, have $q^{2n_1} \geq C$ for $n_1 \in \mathbb{N}$ the number of variables in the initial layer and $C \in \mathbb{N}$ the minimal workload for an attacker, *i.e.*, the security parameter of the system. This is due to the low-rank attack from Section 4.4.2 which directly depends on the number of variables in this layer. In addition, they guard their final layer, *i.e.*, have $q^{m_L} \geq C$ for $m_L \in \mathbb{N}$ the number of new equations in their last layer $L \in \mathbb{N}$. The latter is due to the high-rank attack from Section 4.4.2.

Due to these two attacks, all constructions based on STS must respect this construction principle. In fact, enTTS, TRMS, and Rainbow only differ in the trapdoor they use to this aim: enTTS and Rainbow only use the UOV trapdoor, while TRMS also uses ideas from the mixed field class. In fact, they can use this additional structure to speed up the signature generation time.

Until now, no secure encryption schemes based on the STS class are known. This is similar to the mixed field class which also knows only signature schemes.

6.3 Applications

As we saw in the previous sections, multivariate quadratic schemes have rather large public keys in the range of 8 KByte − 112 KByte. The private key can be smaller, *e.g.*, down to 512 byte in UOV. In terms of signature or message sizes, we can go down until 128 bits (Quartz). In any case, signature verification and encryption take less than 1 ms on a Pentium II with 500 MHz [CGP02]. For Quartz we obtain numbers in the range of 10 s on the same machine [CGP01]. Other schemes report up to 100 ms [YC04b]. Hence, the strong points of multivariate quadratic schemes are short signatures, low message overhead/short signature sizes and fast encryption/signature verification.

Using the observations stated above, we outline application domains which could profit from $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes and develop specific instances of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic signature schemes which can be used in this context. All proposals in this section have an expected security level of 2^{80} — based on our current knowledge of cryptanalysis. A level of 2^{80} 3-DES computations has been identified in the European project [NES] as an adequate security level for nowadays cryptographic applications; this section has been previously published in [WP05a].

6.3.1 Electronic Stamps

The idea here is to replace the current stamping machines by digitally signed stamps which can then be printed on any normal printer — if they are printed more than once, the person who has bought the stamp will be caught, cf [NS00, PV00] for a thorough discussion of this idea. In a nutshell, we have two objectives in this context. First, we want the corresponding signature to be as short as possible — for example, using message recovery techniques, cf [MvOV96]. Second, the signature verification time should be low as the postal service has to verify the signed stamps at a rather high rate.

The characteristics of our proposal are summarised in Table 6.6. We base our proposal on Sflash^{v3} as this is a bijection and hence, we will be able to obtain a valid signature in any case. The overall idea is to compute a 160-bit hash of the whole message, using a cryptographically secure hash function. The remaining 392-160=232 bits are used to encode a part of the message to sign. Hence, the

Table 6.6: Proposed scheme for electronic stamps

Hash [bit]	Parameter	Priv. Key [KByte]	Pub. Key [KByte]	Sign [ms]	Verify [ms]	Expansion [bit]
160	$q = 128$ $n = 67$ $r = 11$	7.8	112.3	< 1	< 1	237

overall message expansion becomes 77 + 160 = 237 bits although the whole signature has — strictly speaking — a size of 469 bits (cf Section 6.1.1 for details on Sflash^{v3}).

6.3.2 Product Activation Keys

For product activation keys, nowadays mostly symmetric key techniques are used. To the knowledge of the author, the idea to use public key techniques for this problem is due to [Ber03]. In contrast to symmetric key techniques, crackers cannot retrieve the symmetric key and hence, they are not able to compute valid activation keys — even if they manage to get a copy of the (public) key of the corresponding product. Therefore, techniques based on asymmetric cryptology are clearly superior — if they allow similar size and speed as their symmetric counterparts. In this thesis, we propose to use a construction based on HFE- as outlined in [CGP01] and with the tweaks proposed in [WP04], cf Section 6.1.2. In particular, we suggest to compute an 80-bit hash from a user-ID of 20/40 bits.

Table 6.7: Proposed schemes for product activation keys

User-ID [bit]	Key [char]	Parameter	Priv. Key [byte]	Pub. Key [KByte]	Gen. [s]	Ver. [ms]	Signature [bit]
20	21	$q = 2$ n=107 $r = 7$	3264	71	≈ 10	< 1	107
40	25	$q = 2$ n=127 $r = 7$	4509	119	≈ 15	< 2	127

The product activation key is then the signature of the 100/120 bits concatenation of the user-ID and the corresponding hash. In symbols: $m := i \mid\mid h(i)$ where m is the 100/120 bit message to be signed, i the 20/40-bit user-id, $h(\cdot)$ a cryptographically secure hash function and $\cdot||\cdot$ the concatenation of bit-strings. In this context we want to point out that this proposal is not vulnerable to the birthday paradox and hence, we do not need a hash-length of 160 bits to achieve a security level of 2^{80}. Therefore, we can avoid the use of the Chained Patarin Construction (cf Section 6.1.2) here. In order to distinguish different products, we suggest to use different public (and hence private) keys for each product as this rules out attacks using valid signatures for one product for another product. In particular, if these products have different prices, it would be an interesting attack to "reuse" the activation key for the cheapest product on the more expensive ones. We want to stress that a public key size in the suggested range is not a problem to be put on a product CD/DVD and hence the additional memory requirement is negligible. Finally, we give the length of the corresponding activation key in characters, assuming a code with 36 symbols. For information: Microsoft uses a 25 character code for its products. The verification and signature timings are extrapolations from [CGP01].

6.3.3 Fast One-Way functions

The last application we see are fast but secure one-way functions. In this case, we do not need a trapdoor but merely the intractability of the $\mathcal{MQ}$-problem. Hence, we suggest to generate random $\mathcal{MQ}$-polynomials with the parameters as suggested in Table 6.8. As for Table 6.6, the evaluation timings are based

Table 6.8: Proposed schemes for one-way functions

Seed [bit]	Parameter	$\mathcal{MQ}$-System [KByte]	Evaluation [ms]
259	$q = 128$, $n = 37$	23	< 1
469	$q = 128$, $n = 67$	134	< 1

on [CGP03a]. A similar construction — but based on sparse polynomials over large finite fields — has been used by Purdy in [Pur74] to construct a kind of hash function. While this proposal is based on the intractability of univariate polynomial equations of large degree, our proposal is based on the difficulty of solving polynomial-equations of small degree, but with a high number of variables. Although the construction we propose here is difficult to invert, it is not resistant against collisions. The reason is a general attack from [Pat96b, Sect. 3, "Attack with related messages"] against $\mathcal{MQ}$-schemes which can be applied here. To counter this attack, we would need to use equations of degree 3 rather than degree 2 — which gives us a larger $\mathcal{MQ}$-system: as we saw in Section 2.2, the size of the public key grows in $O(n^{d+1})$ for $n \in \mathbb{N}$ the number of variables and $d \in \mathbb{N}$ the overall degree. Hence, for quadratic systems we obtain $O(n^3)$ while cubic systems grow with $O(n^4)$. So instead of around 10 kByte, we expect a value in the range of 1 MByte for the public key. Obviously, the public key evaluation for a given input $x \in \mathbb{F}^n$ time will increase accordingly.

6.4 New Schemes and Open Questions

Using the taxonomy developed in this thesis, we are able to derive new schemes — not previously considered in other publications. In particular, we want to stress that mixed schemes should be kept rather simple, so it is possible to determine the strength of the underlying trapdoors. As an overall lesson from the schemes known so far we want to point out that a larger q seems to allow smaller public keys: we have 71 KByte for the public key of Quartz with $q = 2$ in comparison to 15.4 KByte for Sflash^{v2} with q=128, and 8.7 KByte for enhanced TTS and tractable rational signatures with $q = 256$. The reasons for this at first glance rather strange behaviour are the following: a large field size q to decreases the number of variables. But given that the public key is a function of $O(n^3 \log_2 q)$, we see that decreasing n in contrast to q allows us to construct schemes with smaller public keys. However, we cannot do this endlessly: having a very large q, we would obtain $n = 1$ and hence, are in the univariate rather than the multivariate case. Therefore, a choice of $q = 256 = 2^8$ or $q = 65536 = 2^{16}$ seems reasonable at present. After these initial considerations, we now move on to some concrete examples of new schemes.

We are not aware of any successful constructions using variations of UOV or STS. However, STS- may be worthwhile as the minus modification could make the rank attacks difficult. On the other hand, STSi is certainly not a good idea as this boils down increasing the number of variables in the first layer of STS, *i.e.*, STS and STSi are actually the same scheme. We can draw similar conclusions for UOV. As an overall result, we see that more research in this area may be worthwhile. We now outline some schemes which we expect to be worth further research.

6.4.1 MIO

When looking at the taxonomy developed in Section 3.1.5, we see that three of the four schemes, namely HFE, STS, and UOV do allow — at least in principle — odd characteristics. The situation is fundamentally different for MIA: by construction, it only allows even characteristic as the equation $\gcd(q^n - 1, q^\lambda + 1) = 1$ does not have any solution $\lambda \in \mathbb{N}$ otherwise for given $q, n \in \mathbb{N}$ with odd q. In order to ensure that we have a full list of *all* possible schemes, we develop a version of MIA which also works for odd characteristic, called "Matsumoto-Imai odd" (MIO).

As outlined before, we cannot expect any solution for $\gcd(q^n - 1, q^\lambda + 1) = 1$; the closest we get is $\gcd(q^n - 1, q^\lambda + 1) = 2$. Hence, the inversion step in MIO consists of two parts:

1. Using an h such that $h \cdot (q^\lambda + 1) \equiv 2 \pmod{q^n - 1}$ we compute $A := (Y')^h = (X')^2$ in the extension field $\mathbb{E}$

2. Using a general root finding algorithm, we solve the equation $(X')^2 = A$ for given $A \in \mathbb{E}$ and unknown X' (cf Section 3.1.4)

The advantage of such a scheme lies in the fact that root finding becomes more difficult with the degree of the polynomial. Having a degree of 2, the corresponding algorithm will be rather efficient. In contrast to MIA, MIO may not be so efficient as finite fields of even characteristic are particularly well suited for microprocessors.

From a cryptanalytic point of view, MIO offers a few minor advantages over MIA. In particular, the cryptanalysis of [Pat95] is no longer applicable as this paper needs that the scheme in question is a bijection. However, the techniques developed in [FJ03] are certainly applicable. Hence, MIO is not stronger than MIA. But from a mathematical point of view, it is satisfying to have a complete list of all possible schemes, therefore, we decided to include MIO in this chapter.

6.4.2 STS⊥h

With this construction, we want to test the limits of the STS idea as used already in constructions for mixed schemes (cf sections 6.2.1 and 6.2.2). In particular, we want to see which kind of parameters we can use for secure constructions to obtain a lower limit on the public key sizes for schemes of this kind.

We start with noticing that the enhanced TTS class from Section 6.2.1 used linear terms for the new variables of the medium layer and hence, always got a solution here regardless of the input. Similar, the tractable rational map class uses the same trick to ensure that we always obtain a signature for any input. We can sum up this trick under the "⊥" modifier: each equation is independent from all other equations and hence, we can compute the results for one variable independently from all other variables. Next, we recall that the linear and constant terms do not give us any gain in the security of the corresponding scheme. Therefore, in order to obtain smaller public keys, we should avoid them. Actually, this idea has been outlined in the "h" modifier. Finally, STS schemes can be attacked quite successfully both from the highest and the lowest layer, each time using the rank. Hence, a minimal scheme would only use two layers: one with a small rank big enough not to allow any attack here and one with a big rank big enough not to allow any attack from this side.

Remark 6.4.1 *Obviously, we can use a scheme which uses only one layer. However, we are then in the class UOV (cf Section 6.4.3 for a version with secure parameters).*

Hence, the scheme we propose in this section has the following structure for its two layers. We use the notation $a \in \mathbb{N}$ for the input variables for the quadratic polynomials of layer 1, $\alpha \in \mathbb{N}$ for the linear variables of layer 1. Similar, we denote with $b \in \mathbb{N}$ the input of the quadratic polynomials of layer 2 and with $\beta \in \mathbb{N}$ the linear variables of layer 2. Hence, we have $b = a + \alpha$, the number of equations is $m = \alpha + \beta$ and the number of variables is $n = b + \beta$, cf Figure 6.5. Here, we have π_i for $1 \leq i \leq m$ being homogeneous degree 2 polynomials with

$$\begin{array}{rcl}
p'_1(x'_1, \ldots, x'_b) & := & \pi_1(x'_1, \ldots, x'_a) + x'_1 x'_{a+1} \\
p'_2(x'_1, \ldots, x'_b) & := & \pi_2(x'_1, \ldots, x'_a) + x'_1 x'_{a+2} \\
 & \vdots & \\
p'_\alpha(x'_1, \ldots, x'_b) & := & \pi_b(x'_1, \ldots, x'_a) + x'_1 x'_{a+\alpha} \\
p'_{\alpha+1}(x'_1, \ldots, x'_n) & := & \pi_{b+1}(x'_1, \ldots, x'_b) + x'_1 x'_{b+1} \\
p'_{\alpha+2}(x'_1, \ldots, x'_n) & := & \pi_{b+2}(x'_1, \ldots, x'_b) + x'_1 x'_{b+2} \\
 & \vdots & \\
p'_m(x'_1, \ldots, x'_n) & := & \pi_m(x'_1, \ldots, x'_b) + x'_1 x'_{b+\beta}
\end{array}$$

Figure 6.5: STS⊥h with Two Layers

random coefficients. Therefore, all polynomials p'_i are homogeneous degree 2 polynomials. Hence, using $S \in \mathrm{Hom}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Hom}^{-1}(\mathbb{F}^m)$ for the two transformations we obtain a public key which does not contain any linear or constant terms. So the "homogenising modification" has already been built into the trapdoor used. Moreover, the first layer can be inverted by assigning random values to the variables $x'_1, \ldots, x'_a$ as we saw it, *e.g.*, for UOV (cf Section 3.1.1).

There are three important attacks we have to take into account for this scheme: first, we need to make sure that the low rank attacks do not apply and hence, we need $q^{2a} \geq C$ for some security parameter C: as we saw in Section 4.4, the corresponding attack is a function of the rank of the corresponding matrix. Inspecting the matrix in Figure 4.2 we see that this rank grows with $2a$ for sparse private key polynomials of the above form. Second, we need the high rank attacks to be inefficient. Therefore, we obtain $2^{q^{n-b}} = 2^\beta \geq C$, again using the cryptanalysis of Section 4.4. Finally, we need to make sure that the overall construction does not fall to the attacks for schemes from the UOV class, *i.e.*, we need $q^{b-\beta-1} \geq C$. In all cases, we omitted polynomials for the corresponding attacks as we are more interested in the overall asymptotic complexity rather than a "close match". Moreover, we use a security bound of $C = 2^{80}$ for the following constructions. Such a security bound has been suggested, *e.g.*, in the European NESSIE project [NES]. To our knowledge, the follow-up project Ecrypt did not raise this security requirement until now.

Now, with $q = 256 = 2^8$ the following sets of parameters allows a secure construction: $a = 5, \alpha = 16$, *i.e.*, $b = 21$. Moreover, we choose $\beta = 10$ and obtain a total of $n = 31$ variables and $m = 26$ equations. This translates to a public key size of 12,896 bytes. As we see, this is worse than the parameters used in enhanced TTS or rational tractable maps. However, these construction use more than two layers and hence, obtain a higher number of quadratic variables for the last layer. Therefore, attacks using the UOV structure of this construction are less efficient. Similar, the Rainbow construction uses several layers and hence, allows less variables in the final layer. In any case: the other attacks outlined in this thesis did not prove efficient against this kind of schemes and are hence omitted from the above security analysis.

Finally, we also give the parameters for $q = 65536 = 2^{16}$. Here we obtain $a = 3, \alpha = 8$, *i.e.*, $b = 11$ and $\beta = 5$ and obtain a total of $n = 16$ variables and $m = 13$ equations. This translates to a public key size of 3536 bytes. As we saw, the previously mentioned rational to choose q rather large helped up obtaining a (far) smaller public key. However, as we need operations over GF(2^{16}) now, the corresponding scheme may be less suited for smart card implementations as low-end cards still widely use 8-bit microprocessors. However, [WHL+05] discusses some tricks to use "towers of fields" like GF(2^{16}) = GF($(2^8)^2$) in this context. Moreover, we have to take the running time of Gröbner algorithms into account now. Unfortunately, we are not aware of a systematic study of the exact behaviour of Gröbner attacks and hence, have to leave the security of the parameters proposed here as an open problem.

Using sparse polynomials for π_i with $1 \leq i \leq m$ would allow faster generation of the public key and also faster inversion. However, generating secure sparse polynomials is outside of the scope of this thesis. Still, we believe that such a modification would allow a more efficient scheme.

6.4.3 UOV⊥h

The starting point of this construction are [YC04a] and [DS05b]. In a nutshell, they solve the problem of UOV that not all tries in the private key yield a valid signature. They do so by forcing a special matrix structure on the oil variables: no matter which values we choose for the vinegar variables, the oil variables always yield a matrix of full rank, and hence, we can always compute a solution. This can be summarised under the $\perp$ idea (cf previous section).

Here, we use this idea with a slight twist, *i.e.*, with the homogenising modifier. We construct the UOV trapdoor as shown in Figure 6.6 having $o = m$ and $n =$

$$\begin{array}{rcl} p'_1(x'_1,\ldots,x'_n) & := & \pi(x'_1,\ldots,x'_v) + x'_1x'_{v+1} \\ p'_2(x'_1,\ldots,x'_n) & := & \pi(x'_1,\ldots,x'_v) + x'_1x'_{v+2} \\ & \vdots & \\ p'_m(x'_1,\ldots,x'_n) & := & \pi(x'_1,\ldots,x'_v) + x'_1x'_{v+o} \end{array}$$

Figure 6.6: UOV with Branching and Homogenising Modifiers

$v+o$ (cf Section 3.1.1). By choosing $x'_1 \in_R \mathbb{F}^*$, *i.e.*, non-zero, we always obtain a valid signature. Moreover, if we choose the polynomials $\pi_1, \ldots, \pi_m$ homogeneous of degree 2, we obtain a central map $\mathcal{P}'$ which is also homogeneous of degree 2. So, having the two transformations $S \in \mathrm{Hom}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Hom}^{-1}(\mathbb{F}^m)$ linear rather than affine, we hide this internal structure using the T-transformation and do not introduce any linear terms with the S-transformation. As a consequence, the public key does not have linear or constant terms and hence, we save $m(n+1)$ coefficients in total. The overall scheme does still have the same security as UOV, *i.e.*, all known attacks apply and we need to choose the parameter accordingly.

In any case, multiplying the monomials $x'_1x'v + i$ for $i = 1 \ldots o$ with random coefficients $\gamma'_{i,1,v+i} \in_R \mathbb{F}^*$ does not improve the security of UOV⊥h: applying the ideas developed in [WP05c] we see that such coefficients would lead to equivalent keys and are hence a waste of memory.

Possible parameters for this scheme over GF(256) are $m = o = 20$ and $v = 40$. This leads to $n = 60$ and therefore to a public key of length 36,000 bytes. Over GF(2^{16}), a choice of $m = o = 10$ and $v = 20$ would be secure, assuming the attacker cannot compute 2^{80} 3-DES operations In particular, this leads to $n = 30$ variables and an overall public key of 8300 bytes.

As for any other scheme, choosing the vinegar polynomials $\pi_1, \ldots, \pi_m$ sparse rather than dense allows a speed-up.

STS-

A quite interesting but recent development are schemes of the STS- class, as suggested in [KS05]. The overall construction described there is based on the (broken) RSE(2)PKC and RSSE(2)PKC (cf Section 4.4.5 for details).

In short, they suggest the use of the STS class together with the reduction transformation (cf Section 2.4) for details. It is interesting to notice that the rank attacks described in Section 4.4 are in vain here: in all cases, they need to reconstruct the affine transformation T first. However, as parts of this transformation are missing now, this is no longer uniquely possible. On the other hand, the initial birational permutation scheme of Shamir [Sha93] already used the minus modification and was broken in [CSV97] nevertheless. It would be interesting to see if this cryptanalysis was possible due to some special properties of birational permutations or if there are intrinsic properties of the STS class which make the minus modification non-effective.

If the latter is not the case, STS- could be used in form of regular STS with a field size of $q = 256$ and a only $r = 1$ new variables on each layer plus $r' = 1$ one equation missing in the reduction transformation. Obviously, such a scheme could be used for encryption. Still, a thorough security analysis is imperative before using it.

6.5 Discussion

In this thesis, we gave a concise overview of an alternative class of public key schemes, called "$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic" schemes. In particular — using the variations HFE- and MIA- — we developed practical instantiations for the problems of fast one-way functions, electronic stamps, and product activation keys. In all cases, the short signature verification times and also the rather short signature generation times (resp., encryption and decryption) are a clear advantage over schemes based on RSA and ECC. In particular, the author is not aware of patent-restrictions for HFE- and MIA-. The situation is different for HFEv, where Axalto (former Schlumberger) seems to hold a patent [CGP01]. Hence, HFE- and MIA- are a good alternative for projects where patent royalties are a serious consideration. We also want to point out that the predecessor of Sflash^{v3}, *i.e.*, Sflash^{v2} has been recommended by NESSIE for special application domains. Similar, Quartz was a recommendation in NESSIE for applications which require particularly short signatures.

In addition, we were able to develop special variations in this thesis, namely UOV⊥h and STS⊥h. Both are kind of "tweaked" in comparison to their original versions: in both cases, we always obtain a valid signature and hence, are not forced to repeat the signing process. Moreover, the structure of the private key allows for fast inversion and hence, signature generation. Still, we know from Chapter 5 that these two modification do not allow attacks. Finally, we get rather small public keys, especially for the version over GF(2^{16}). An interesting research problem in this context is the STS- class: if secure, it would allow efficient encryption schemes based on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials. Still, the chances are rather slim as pointed out in the previous section.

Chapter 7

Conclusions

In this thesis, we described the state of the art of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key cryptography. We saw that all schemes known so far fit into an easy taxonomy of only four basic classes, namely unbalanced oil and vinegar (UOV), stepwise triangular systems (STS), Matsumoto-Imai Scheme A (MIA), and hidden field equations (HFE). With this taxonomy it was possible to spot a missing scheme, namely the version for odd characteristic of MIA, denoted MIO. Moreover, this taxonomy allows to put existing schemes in perspective and hence helps developing new schemes.

Apart from these four basic classes, we also found ten basic modifiers, cf Table 3.1 for an overview. Using both the basic classes and the modifiers, we are actually able to express *any* $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic scheme in the terminology developed in this thesis. We demonstrated this in Chapter 6 where we showed that the group around Patarin mainly concentrated on schemes from the mixed field class (HFE and MIA), while the new schemes from Ding (Rainbow), Chen/Yang (enTTS), and Wang (TRMS) all belong to the class of single field schemes and are actually instances of the STS class. This shows that any breakthrough in the cryptanalysis of the STS class will defeat these new schemes. Hence, cryptanalytic work in this area is certainly worthwhile.

Moreover, we saw that all $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic signature schemes suffer from the same drawbacks, *i.e.*, rather large public keys in the range of 8 kByte (*e.g.*, enTTS) up to 71 kByte (*e.g.*, Quartz); all these key-sizes are for a security level of 2^{80} 3-DES computations, cf Section 6 for more details on their performance. As reducing the number of variables makes certain types of attacks easier, we do not expect that the key size decreases significantly in the near future. In particular as constructions like the subfield modification "$\perp$" were broken in the cases of MIA and UOV (cf Chapter 4). Although there is no general result showing that the "$\perp$" modification is insecure, we do not expect that it can be safely used with any practical scheme. In addition, we saw that $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic constructions only allows secure signature schemes so far. Still, signature generation is generally fast and can be computed, *e.g.*, on a low-end smart-card without a cryptographic co-processor. This makes $\mathcal{MQ}$-schemes an interesting choice for restricted environments and further research in this direction is certainly worthwhile. Taking their disadvantages and advantages into account, we were able to point to possible application domains such as electronic stamps, product activation keys and fast one-way functions. In each case, the disadvantages of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems are outweighed by their advantages.

From a mathematical point of view, the question of equivalent keys (cf Chapter 5) proved interesting. An important property in this context is that equivalent keys allow to reduce the memory requirements for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems without jeopardising their security. As we see the main application domain of $\mathcal{MQ}$-systems in (memory) restricted environments, this result has practical applications — although it was derived from a purely theoretical point of view, namely studying the structure of the key spaces of $\mathcal{MQ}$-schemes. An additional benefit of these results can be a lower computational requirement for some $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems: taking equivalent keys into account, we can compute a normal form of a given private key. This normal form has a very specific form which can be exploited in a fast implementation. One example is using the fact that many terms have to be multiplied with a constant of 1 or 0, *i.e.*, we are able to omit this multiplication and sometimes even the corresponding term.

Hence, $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems are an interesting and worthwhile research topic. At present, not as much work as for RSA or ECC has been done. On the down-side, this means that the security of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic systems is not as well understood as this is the case for other schemes. On the up-side, this indicates that there are still many open questions. In particular, finding better attacks for some of the basic schemes or basic schemes combined with some modifiers is worthwhile. A very interesting topic in this context is the construction of an encryption scheme based on $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic polynomials. Until now, there is no practical solution for this primitive. The only exception so far is HFEi from Ding. However, as his construction MIAi was broken shortly after its publication, it seems reasonable to wait a while until using HFEi. Given that the whole area of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems has not been investigated deeply by many independent researchers, it seems reasonable to wait 5–10 years before using them in practical applications. However, for low-security applications with a clearly defined, small risk, they can already be an option. All in all, we hope that the subject of $\mathcal{MQ}$-systems will attract more attention over the coming years.

Bibliography

[AFI+04] Gwénolé Ars, Jean-Charles Faugère, Hideki Imai, Mitsuru Kawazoe, and Makoto Sugita. Comparison between xl and gröbner basis algorithms. In *Advances in Cryptology — ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 338–353. Pil Joong Lee, editor, Springer, 2004.

[Bau95] Friedrich L. Bauer. *Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie*. Springer, 1995. ISBN 3-540-58118-9.

[Ber03] Giuliano Bertoletti. Private communication, June 2003.

[Beu93] Albrecht Beutelspacher. *Kryptologie*. Vieweg, 3rd edition, 1993. ISBN 3-528-28990.

[BFS96] Jonathan F. Buss, Gudmund Skovbjerg Frandsen, and Jeffrey Outlaw Shallit. The computational complexity of some problems of linear algebra. Research Series RS-96-33, BRICS, Department of Computer Science, University of Aarhus, September 1996. `http://www.brics.dk/RS/96/33/`, 39 pages.

[BSS99] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999. ISBN 0-521-65374-6.

[BWP05] An Braeken, Christopher Wolf, and Bart Preneel. A study of the security of Unbalanced Oil and Vinegar signature schemes. In *The Cryptographer's Track at RSA Conference 2005*, volume 3376 of *Lecture Notes in Computer Science*. Alfred J. Menezes, editor, Springer, 2005. 13 pages, cf `http://eprint.iacr.org/2004/222/`.

[CDF03] Nicolas T. Courtois, Magnus Daum, and Patrick Felke. On the security of HFE, HFEv- and Quartz. In *Public Key Cryptography — PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 337–350. Y. Desmedt, editor, Springer, 2002. `http://eprint.iacr.org/2002/138`.

[CGMT02] Nicolas Courtois, Louis Goubin, Willi Meier, and Jean-Daniel Tacier. Solving underdefined systems of multivariate quadratic equations. In *Public Key Cryptography — PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 211–227. David Naccache and Pascal Paillier, editors, Springer, 2002.

[CGP00a] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *Flash: Primitive specification and supporting documentation*, 2000. `https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/flash.zip`, 9 pages.

[CGP00b] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *Quartz: Primitive specification and supporting documentation*, 2000. `https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/quar%tz.zip`, 15 pages.

[CGP00c] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash: Primitive specification and supporting documentation*, 2000. `https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/sfla%sh.zip`, 10 pages.

[CGP01] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *Quartz: Primitive specification (second revised version)*, October 2001. `https://www.cosic.esat.kuleuven.ac.be/nessie` Submissions, Quartz, 18 pages.

[CGP02] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash: Primitive specification (second revised version)*, 2002. `https://www.cosic.esat.kuleuven.ac.be/nessie`, Submissions, Sflash, 11 pages.

[CGP03a] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash^{v3}, a fast asymmetric signature scheme — Revised Specificatoin of SFlash, version 3.0*, October 17$^{\text{th}}$ 2003. ePrint Report 2003/211, `http://eprint.iacr.org/`, 14 pages.

[CGP03b] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash^{v3}, a fast asymmetric signature scheme — Revised Specificatoin of SFlash, version 3.0*, October 2$^{\text{nd}}$ 2003. ePrint Report 2003/211, `http://eprint.iacr.org/`, 13 pages.

[CKPS00] Nicolas T. Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Bart Preneel, editor, Springer, 2000. Extended Version: `http://www.minrank.org/xlfull.pdf`.

[Cou01] Nicolas T. Courtois. The security of Hidden Field Equations (HFE). In *The Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 266–281. D. Naccache, editor, Springer, 2001. `http://www.minrank.org/hfesec.{ps|dvi|pdf}`.

[Cou04] Nicolas Courtois. Algebraic attacks over gf(2^{k}), application to HFE challenge 2 and Sflash-v2. In PKC [PKC04], pages 201–217. ISBN 3-540-21018-0.

[Cr93] Douglas R. Stinson, editor. *Advances in Cryptology — CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*. Springer, 1993. ISBN 3-540-57766-1.

[Cr95] Don Coppersmith, editor. *Advances in Cryptology — CRYPTO 1995*, volume 963 of *Lecture Notes in Computer Science*. Springer, 1995. ISBN 3-540-60221-6.

[CSV93] Don Coppersmith, Jacques Stern, and Serge Vaudenay. Attacks on the birational permutation signature schemes. In Cr [Cr93], pages 435–443.

[CSV97] Don Coppersmith, Jacques Stern, and Serge Vaudenay. The security of the birational permutation signature schemes. *Jounal of Cryptology*, 10:207–221, 1997.

[DA99] T. Dierks and C. Allen. *RFC 2246: The TLS Protocol (Version 1.0)*. Internet Society, January 1999. `http://www.rfc-editor.org`, 80 pages.

[Dau01] Magnus Daum. Das Kryptosystem HFE und quadratische Gleichungssysteme über endlichen Körpern. Diplomarbeit, Universität Dortmund, August 2001. `http://homepage.ruhr-uni-bochum.de/Magnus.Daum/HFE.{ps.zip|pdf}`, 133 pages.

[DF05] Hans Dobbertin and Patrick Felke. Mistery twister, cryptochallenge 11. `http://www.mistery-twister.com/`, 8 pages, April 2005.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[Din04] Jintai Ding. A new variant of the matsumoto-imai cryptosystem through perturbation. In PKC [PKC04], pages 305–318.

[DS04] Jintai Ding and Dieter Schmidt. Multivariable public-key cryptosystems. Preprint, 16^{th} of December 2004. 16 pages.

[DS05a] Jintai Ding and Dieter Schmidt. Cryptanalysis of HFEv and internal perturbation of HFE. In PKC [PKC05], pages 288–301.

[DS05b] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *Conference on Applied Cryptography and Network Security — ACNS 2005, to appear in LNCS*. Springer, 2005. 10 pages.

[DY04] Jintai Ding and Zhijun Yin. Cryptanalysis of TTS and Tame-like multivariate signature schemes. Pre-Proceedings of the The Third International Workshop for Applied PKI, Fukuoka, Japan, October 3-5., 2004.

[ECr05] Ronald Cramer, editor. *Advances in Cryptology — EUROCRYPT 2005*, Lecture Notes in Computer Science. Springer, 2005.

[Fau02] Jean-Charles Faugère. HFE challenge 1 broken in 96 hours. Announcement that appeared in `news://sci.crypt`, 19^{th} of April 2002.

[Fau03] Jean-Charles Faugère. Algebraic cryptanalysis of (HFE) using Gröbner bases. Technical report, Institut National de Recherche en Informatique et en Automatique, February 2003. `http://www.inria.fr/rrrt/rr-4738.html`, 19 pages.

[FC 00] Yair Frankel, editor. *Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings*, volume 1962 of *Lecture Notes in Computer Science*. Springer, 2001. ISBN 3-540-42700-7.

[FD85] Harriet Fell and Whitfield Diffie. Analysis of public key approach based on polynomial substitution. In *Advances in Cryptology — CRYPTO 1985*, volume 218 of *Lecture Notes in Computer Science*, pages 340–349. Hugh C. Williams, editor, Springer, 1985.

[Fel01] Patrick Felke. Multivariate Kryptosysteme, insbesondere das Schema von Imai und Matsumoto. Diplomarbeit, Universität Dortmund, August 2001. 146 pages.

[Fel04] Patrick Felke. On the affine transformations of HFE-cryptosystems and systems with branches. Cryptology ePrint Archive, Report 2004/367, 2004. `http://eprint.iacr.org/2004/367`, version from 2004-12-17, 10 pages.

[FGS05] Pierre-Alain Fouque, Louis Granboulan, and Jacques Stern. Differential cryptanalysis for multivariate schemes. In ECr [ECr05]. 341–353.

[FIP] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 180-1: Secure Hash Standard*, 17^{th} April 1995. `http://www.itl.nist.gov/fipspubs/fip180-1.htm`.

[FIP01] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 180-2: Secure Hash Standard*, August 2001. `http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf`.

[FJ03] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of Hidden Field Equations (HFE) using gröbner bases. In *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Dan Boneh, editor, Springer, 2003.

[FY79] A.S. Fraenkel and Y. Yesha. Complexity of problems in games, graphs and algebraic equations. *Discrete Applied Mathematics*, 1(1-2):15–30, September 1979.

[GC00] Louis Goubin and Nicolas T. Courtois. Cryptanalysis of the TTM cryptosystem. In *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 44–57. Tatsuaki Okamoto, editor, Springer, 2000.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979. ISBN 0-7167-1044-7 or 0-7167-1045-5.

[GM02] Henri Gilbert and Marine Minier. Cryptanalysis of SFLASH. In *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 288–298. Lars R. Knudsen, editor, Springer, 2002.

[GMS02] Willi Geiselmann, Willi Meier, and Rainer Steinwandt. An attack on the Isomorphisms of Polynomials problem with one secret. Cryptology ePrint

Archive, Report 2002/143, 2002. `http://eprint.iacr.org/2002/143`, version from 2002-09-20, 12 pages.

[Gra05] Louis Granboulan. A generic scheme based on trapdoor one-way permutations with signatures as short as possible. In PKC [PKC05], pages 302–312.

[GSB01] W. Geiselmann, R. Steinwandt, and Th. Beth. Attacking the affine parts of SFlash. In *Cryptography and Coding - 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 355–359. B. Honary, editor, Springer, 2001. Extended version: `http://eprint.iacr.org/2003/220/`.

[IM85] Hideki Imai and Tsutomu Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In *Algebraic Algorithms and Error-Correcting Codes, 3rd International Conference, AAECC-3, Grenoble, France, July 15-19, 1985, Proceedings*, volume 229 of *Lecture Notes in Computer Science*, pages 108–119. Jacques Calmet, editor, Springer, 1985.

[JKJMR05] Antoine Joux, Sébastien Kunz-Jacques, Frédéric Muller, and Pierre-Michel Ricordel. Cryptanalysis of the tractable rational map cryptosystem. In PKC [PKC05], pages 258–274.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In *Advances in Cryptology — EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222. Jacques Stern, editor, Springer, 1999.

[KPG03] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes — extended version, 2003. 17 pages, `citeseer/231623.html`, 2003-06-11.

[KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil and vinegar signature scheme. In *Advances in Cryptology — CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 257–266. Hugo Krawczyk, editor, Springer, 1998.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem. In *Advances in Cryptology — CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Michael Wiener, editor, Springer, 1999. `http://www.minrank.org/hfesubreg.ps` or `http://citeseer.nj.nec.com/kipnis99cryptanalysis.html`.

[KS04a] Masao Kasahara and Ryuichi Sakai. private communication, 3rd of April 2004.

[KS04b] Masao Kasahara and Ryuichi Sakai. A construction of public-key cryptosystem based on singular simultaneous equations. In *Symposium on Cryptography and Information Security — SCIS 2004*. The Institute of Electronics, Information and Communication Engineers, January 27–30 2004. 6 pages.

[KS04c] Masao Kasahara and Ryuichi Sakai. A construction of public key cryptosystem for realizing ciphtertext of size 100 bit and digital signature scheme. *IEICE Trans. Fundamentals*, E87-A(1):102–109, January 2004. Electronic version: `http://search.ieice.org/2004/files/e000a01.htm\#e87-a,1,102`.

[KS05] Massao Kasahara and Ryuichi Sakai. A construction of public-key crypto based on singular simultaneous equations and its variants. Technical report, 2005. `http://www.osaka-gu.ac.jp/php/kasahara/km.pdf`, 6 pages.

[LD00] Julio Lopéz and Ricardo Dahab. An overview of elliptic curve cryptography. Technical report, Institute of Computing, State University of Campinas, Brazil, 22^{nd} of May 2000. `http://citeseer.nj.nec.com/333066.html` or `http://www.dcc.unicamp.br/ic-tr-ftp/2000/00-14.ps.gz`.

[LN00] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge University Press, 2000. ISBN 0-521-46094-8.

[LP03] Françoise Levy-dit-Vehel and Ludovic Perret. Polynomial equivalence problems and applications to multivariate cryptosystems. In *Progress in Cryptology — INDOCRYPT 2003*, volume 2904 of *Lecture Notes in Computer Science*, pages 235–251. Thomas Johansson and Subhamoy Maitra, editors, Springer, 2003.

[MAG] Computational Algebra Group, University of Sydney. *The MAGMA Computational Algebra System for Algebra, Number Theory and Geometry*. `http://magma.maths.usyd.edu.au/magma/`.

[Mar01] Gwenaëlle Martinet. Selecting a C^{*--} scheme. Report for [NES], Document NES/DOC/ENS/WP3/009/a, 2001. `https://www.cosic.esat.kuleuven.ac.be/nessie/reports/enswp3-009a.pdf`.

[Mas92] James L. Massey. *Contemporary Cryptology — The Science of Information Integrity*, chapter Contemporary Cryptology — An Introduction, pages 1–39. IEEE, 1992. ISBN 0-87942-277-7.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature verification and message-encryption. In *Advances in Cryptology — EUROCRYPT 1988*, volume 330 of *Lecture Notes in Computer Science*, pages 419–545. Christoph G. Günther, editor, Springer, 1988.

[MIHM85] Tsutomu Matsumoto, Hideki Imai, Hiroshi Harashima, and Hiroshi Miyakawa. A cryptographically useful theorem on the connection between uni and multivariate polynomials. *Transactions of the IECE of Japan*, 68(3):139–146, March 1985.

[Moh99] T. Moh. A public key system with signature and master key function. *Communications in Algebra*, 27(5):2207–2222, 1999. Electronic version: `http://citeseer/moh99public.html`.

[MS91] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier Science Publisher, 1991. ISBN 0-444-85193-3.

[MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN 0-8493-8523-7, online-version: `http://www.cacr.math.uwaterloo.ca/hac/`.

[NES] NESSIE: New European Schemes for Signatures, Integrity, and Encryption. Information Society Technologies programme of the European commission (IST-1999-12324). `http://www.cryptonessie.org/`.

[NS00] David Naccache and Jacques Stern. Signing on a postcard. In FC — Financial Crypto [FC 00], pages 121–135. `http://citeseer.ist.psu.edu/naccache00signing.html`.

[Pat95] Jacques Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In Cr [Cr95], pages 248–261.

[Pat96a] Jacques Patarin. Asymmetric cryptography with a hidden monomial. In *Advances in Cryptology — CRYPTO 1996*, volume 1109 of *Lecture Notes in Computer Science*, pages 45–60. Neal Koblitz, editor, Springer, 1996.

[Pat96b] Jacques Patarin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology — EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Ueli Maurer, editor, Springer, 1996. Extended Version: `http://www.minrank.org/hfe.pdf`.

[Pat97] Jacques Patarin. The oil and vinegar signature scheme. presented at the Dagstuhl Workshop on Cryptography, September 1997. transparencies.

[Pat00] Jacques Patarin. Secret public key schemes. In *Public-Key Cryptography and Computational Number Theory 2000*, pages 221–237. Stefan Banach, editor, de Gruyter, 2000.

[PBO$^+$02] B. Preneel, A. Bosselaers, S.B. Ors, A. Biryukov, L. Granboulan, E.Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, P. Serf, E. Eiham, E. Barkan, O. Dunkelman, M. Ciet, F. Sieca, L. Knudsen, and H. Raddum. Update on the selection of algorithms for further investigation during the second round. Document NES/DOC/ENS/WP5/D18/1, March 2002. `https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D18_1.pdf`, 22 pages.

[PBO$^+$03] B. Preneel, A. Biryukov, E. Oswald, B. Van Rompay, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, M. Schenfheutle, P. Serf, E. Biham, E. Barkan, O. Dunkelman, J.-J. Quisquater, M. Ciet, F. Sica, L. Knudsen, M. Parker, and H. Raddum. NESSIE security report, version 2.0. Document NES/DOC/ENS/WP5/D20/2, 19$^{\text{th}}$ of February 2003. `https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D20-v2.pdf`, see [NES], 342 pages.

[Per05] Ludovic Perret. A fast cryptanalysis of the isomorphism of polynomials with one secret problem. In ECr [ECr05]. 17 pages.

[PG97] Jacques Patarin and Louis Goubin. Trapdoor one-way permutations and multivariate polynomials. In *International Conference on Information Security and Cryptology 1997*, volume 1334 of *Lecture Notes in Computer Science*, pages 356–368. International Communications and Information Security Association, Springer, 1997. Extended Version: `http://citeseer.nj.nec.com/patarin97trapdoor.html`.

[PGC98a] Jacques Patarin, Louis Goubin, and Nicolas Courtois. C^*_{-+} and HM: Variations around two schemes of T.Matsumoto and H.Imai. In *Advances in Cryptology — ASIACRYPT 1998*, volume 1514 of *Lecture Notes in Computer Science*, pages 35–49. Kazuo Ohta and Dingyi Pei, editors, Springer, 1998. Extended Version: `http://citeseer.nj.nec.com/patarin98plusmn.html`.

[PGC98b] Jacques Patarin, Louis Goubin, and Nicolas Courtois. Improved algorithms for Isomorphisms of Polynomials. In *Advances in Cryptology — EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 184–200. Kaisa Nyberg, editor, Springer, 1998. Extended Version: `http://www.minrank.org/ip6long.ps`.

[PKC04] Feng Bao, Robert H. Deng, and Jianying Zhou (editors). *Public Key Cryptography — PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*. Springer, 2004. ISBN 3-540-21018-0.

[PKC05] Serge Vaudenay, editor. *Public Key Cryptography — PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*. Springer, 2005. ISBN 3-540-24454-9.

[Pur74] George B. Purdy. A high security log-in procedure. *Communications of the ACM*, 17(8):442–445, August 1974.

[PV00] Leon A. Pintsov and Scott A Vanstone. Postal revenue collection in the digital age. In FC — Financial Crypto [FC 00], pages 105–120. `http://citeseer.ist.psu.edu/pintsov00postal.html`.

[Sch96] Bruce Schneier. *Applied Cryptography - protocols, algorithms, and source code in C*. John Wiley & Sons, Inc., 2nd edition, 1996. ISBN 0-471-12845-7 or 0-471-11709-9.

[SG03] Andrey V. Sidorenko and Ernst M. Gabidulin. The weak keys for HFE. In *Proceedings of the "Seventh International Symposium on Communication Theory and Applications (ISCTA03)"; 13th-18th July, 2003, Ambleside, Lake District, UK*, pages 239–244, 2003. 6 pages.

[Sha93] Adi Shamir. Efficient signature schemes based on birational permutations. In Cr [Cr93], pages 1–12.

[Shi00] R. Shirey. *RFC 2828: Internet Security Glossary*. Internet Society, May 2000. `http://www.rfc-editor.org`, 212 pages.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[The95] Thorsten Theobald. How to break Shamir's asymmetric basis. In Cr [Cr95], pages 136–147.

[Tol03] Ilia Toli. Cryptanalysis of HFE, June 2003. arXiv preprint server, `http://arxiv.org/abs/cs.CR/0305034`, 7 pages.

[Wan05] Lih-Chung Wang. private communication, 16^{th} of March 2005.

[WBP04] Christopher Wolf, An Braeken, and Bart Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. In *Conference on Security in Communication Networks — SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 294–309. Springer, September 8–10 2004. Extended version: `http://eprint.iacr.org/2004/237`.

[WC04] Lih-Chung Wang and Fei-Hwang Chang. Tractable rational map cryptosystem. Cryptology ePrint Archive, Report 2004/046, 18th of February 2004. `http://eprint.iacr.org/2004/046/`.

[WHL$^+$05] Lih-Chung Wang, Yuh-Hua Hu, Feipei Lai, Chun-Yen Chou, and Bo-Yin Yang. Tractable rational map signature. In PKC [PKC05], pages 244–257.

[Wol02a] Christopher Wolf. *Hidden Field Equations* (HFE) - variations and attacks. Diplomarbeit, Universität Ulm, December 2002. `http://www.christopher-wolf.de/dpl`, 87 pages.

[Wol02b] Christopher Wolf. Implementing Quartz in Java. In *Third Open NESSIE Workshop*, Munich, November 2002. Louis Granboulan, editor. 12 pages.

[Wol04] Christopher Wolf. Efficient public key generation for HFE and variations. In *Cryptographic Algorithms and Their Uses 2004*, pages 78–93. Dawson, Klimm, editors, QUT University, 2004.

[WP04] Christopher Wolf and Bart Preneel. Asymmetric cryptography: Hidden Field Equations. In *European Congress on Computational Methods in Applied Sciences and Engineering 2004*. P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, Jyväskylä University, 2004. 20 pages, extended version: `http://eprint.iacr.org/2004/072/`.

[WP05a] Christopher Wolf and Bart Preneel. Applications of multivariate quadratic public key systems. In *Sicherheit 2005; Sicherheit — Schutz und Zuverlässigkeit*, LNI — Lecture Notes in Informatics, pages 413–424, April 5–8 2005. Extended version `http://eprint.iacr.org/2004/236/`.

[WP05b] Christopher Wolf and Bart Preneel. Equivalent keys in HFE, C^*, and variations. In *Proceedings of Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 33–49. Serge Vaudenay, editor, Springer, 2005. Extended version `http://eprint.iacr.org/2004/360/`, 15 pages.

[WP05c] Christopher Wolf and Bart Preneel. Superfluous keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic asymmetric systems. In PKC [PKC05], pages 275–287. Extended version `http://eprint.iacr.org/2004/361/`.

[WP05d] Christopher Wolf and Bart Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint

Archive, Report 2005/077, 12th of May 2005. `http://eprint.iacr.org/2005/077/`, 64 pages.

[YC04a] Bo-Yin Yang and Jiun-Ming Chen. Rank attacks and defence in Tame-like multivariate PKC's. Cryptology ePrint Archive, Report 2004/061, 23rd February 2004. `http://eprint.iacr.org/`, 17 pages.

[YC04b] Bo-Yin Yang and Jiun-Ming Chen. Rank attacks and defence in Tame-like multivariate PKC's. Cryptology ePrint Archive, Report 2004/061, 29rd September 2004. `http://eprint.iacr.org/`, 21 pages.

[YG01] Amr M. Youssef and Guang Gong. Cryptanalysis of Imai and Matsumoto scheme B asymmetric cryptosystem. In *Progress in Cryptology — INDOCRYPT 2001*, volume 2247 of *Lecture Notes in Computer Science*, pages 214–222. C. Pandu Rangan and Cunsheng Ding, editors, Springer, 2001.

Index

List of Publications

Journals

1. P. Fitzpatrick und C. Wolf: "Direct division in factor rings", Electronic Letters, Vol. 38, No. 21, p. 1253-1254, October 10, 2002.

Editor

1. C. Wolf, St. Lucks, P.-W. Yau (Eds.), "WEWoRC 2005 — Western Europen Workshop on Research in Cryptology, In the Lecture Notes in Informatics (LNI) P-74, Gesellschaft für Informatik, 2005.

International Conferences/Workshops

1. C. Wolf and B. Preneel, "Equivalent Keys for HFE, C^*, and variations", In Mycrypt 2005, Lecture Notes in Computer Science LNCS 3715, Ed Dawson, Serge Vaudenay (Eds.), Springer-Verlag, pp. 33–49, 2005. Springer-Verlag, 2005.

2. A. Braeken, C. Wolf, and B. Preneel, "A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes", In Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference, Lecture Notes in Computer Science LNCS 3376, A. Menezes (ed.), Springer-Verlag, pp. 29–43, 2005.

3. C. Wolf, and B. Preneel, "Superfluous Keys in Multivariate Quadratic Asymmetric Systems", In Public Key Cryptography, 8^{th} International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2005, Lecture Notes in Computer Science 3386, S. Vaudenay (ed.), Springer-Verlag, pp. 275–287, 2005.

4. A. Braeken, C. Wolf, and B. Preneel, "Efficient Cryptanalysis of RSE(2)PKC and RSSE(2)PKC", In Security in Communication Networks, 4^{th} International Conference, SCN 2004, Lecture Notes in Computer Science 3352, C. Blundo, and S. Cimato (eds.), Springer-Verlag, pp. 294-307, 2005.

5. A. Braeken, C. Wolf, and B. Preneel, "A Randomised Algorithm for Checking the Normality of Cryptographic Boolean Functions", In 3rd International Conference on Theoretical Computer Science 2004, J. Levy, E. W. Mayr, and J. C. Mitchell (eds.), Kluwer, pp. 51–66, 2004.

6. C. Wolf, and B. Preneel, "Asymmetric Cryptography: Hidden Field Equations", In European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Jyvaskyla University Press, 20 pages, 2004.

7. C. Wolf, "Overview of Hidden Field Equations", In Polynomial-based Cryptography 2004, 3 pages, 2004.

8. C. Wolf, "Efficient Public Key Generation for HFE and Variations", In Cryptographic Algorithms and their Uses — 2004, E. Dawson, and W. Klemm (eds.), QUT Publications, pp. 78–93, 2004.

9. C. Wolf, "Deriving Public Polynomials Efficiently for HFE-like Systems", In 2nd Yet Another Conference on Cryptography (YACC), 1 pages, 2004.

10. C. Wolf, "Implemeting Quartz in Java, 3rd New European Schemes for Signatures, Integrity, and Encryption Workshop, Munich, Germany, November 6–7, 2002, 12 pages.

Journals (national level)

1. C. Wolf and E. Zenner, "Zur Sicherheit von SHA-1, Tragweite und Konsequenzen — ein aktueller Überblick", Datenschutz und Datensicherheit 29 (5), pp. 275–278, 2005.

Editor (national level)

1. W. Lindner and C. Wolf (Hrsg.), "2. Kryptotag - Workshop ber Kryptographie", Technical Report 2005-CW-1, COSIC, KU Leuven, Belgium and Ulmer Informatik Berichte Nr. 2005-02, University of Ulm, Germany, 12 pages, 2005.

2. S. Lucks and C. Wolf (Hrsg.), "1. Kryptotag - Workshop ber Kryptographie", Technical Report 2004-CW-1, COSIC, KU Leuven, Belgium and TR 2004-10, Reihe Informatik, University of Mannheim, Germany, 17 pages, 2004.

National Conferences/Workshops

1. C. Wolf and B. Preneel, "Applications of Multivariate Quadratic Public Key Systems", In Sicherheit - Schutz und Zuverlässigkeit, Lecture Notes in Informatics (LNI) P-62, Gesellschaft für Informatik, pp. 413–424, 2005.

2. C. Wolf, P. Fitzpatrick, S.N. Foley, und E. Popovici, "HFE in Java: Implementing Hidden Field Equations for Public Key Cryptography", Irish Signals and Systems Conference (ISSC), June 24 - 26, 2002, Cork, Ireland, 6 pages.

Christopher Wolf was born on May 7, 1977 in Bobingen, Germany. He received the degree of Master in Computer Science (Diplom Informatiker, Dipl.-Inf.) from the University of Ulm, Germany, in December 2002. In addition, he spent one year in the Master's Qualifying programme of University College Cork, Ireland, starting in September 2001. His Masters' thesis (Diplomarbeit) dealt with the topic of Hidden Field Equations. In January 2003 he started working in the research group COSIC (Computer Security and Industrial Cryptography) at the Department of Electrical Engineering (ESAT) of the K.U.Leuven.