UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Gustavo Zambonin

On the randomness of Rainbow signatures

Florianópolis
2020

Gustavo Zambonin

On the randomness of Rainbow signatures

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de mestre em Ciência da Computação.
Orientador: Prof. Ricardo Felipe Custódio, Dr.
Coorientador: Prof. Daniel Panario, Dr.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Zambonin, Gustavo
   On the randomness of Rainbow signatures / Gustavo
Zambonin ; orientador, Ricardo Felipe Custódio,
coorientador, Daniel Panario, 2020.
   89 p.

   Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Ciência da Computação, Florianópolis, 2020.

   Inclui referências.

   1. Ciência da Computação. 2. Assinatura digital. 3.
Criptografia pós-quântica. 4. Criptografia baseada em
polinômios multivariados. 5. Rainbow. I. Custódio, Ricardo
Felipe. II. Panario, Daniel. III. Universidade Federal de
Santa Catarina. Programa de Pós-Graduação em Ciência da
Computação. IV. Título.

Gustavo Zambonin

On the randomness of Rainbow signatures

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Jintai Ding, Dr.
University of Cincinnati

Prof^{a}. Lucia Moura, Dra.
University of Ottawa

Prof. Ricardo Dahab, Dr.
Universidade Estadual de Campinas

Prof. Jean Everson Martina, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Ciência da Computação.

Prof^{a}. Vania Bogorny, Dra.
Coordenadora do Programa

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis, 2020.

ACKNOWLEDGEMENTS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e com apoio do programa Mitacs-CALAREO Globalink Research Award (Canadá).

Agradeço aos meus pais, Roberto e Luciana, pelo suporte e confiança depositados em mim ao longo do desenvolvimento deste trabalho. Agradeço às amigas e amigos, Ana Letícia, Douglas Martins, Douglas Silva, Larissa e Matheus, pela companhia na forma de inúmeras conversas após (ou durante) um dia de trabalho, e pela paciência quando estas tornavam-se desabafos.

Agradeço aos Professores Ricardo Custódio, Daniel Panario, Jean Martina, Ricardo Dahab e Jintai Ding, e Professora Lucia Moura, que auxiliaram de várias maneiras no progresso desta pesquisa, através de sua hospitalidade em algumas latitudes (inclusive as virtuais), orientação, disponibilidade e olhar crítico.

Por fim, agradeço à Poliana, pela presença em todo e qualquer um dos dias que compuseram esta jornada, pela sua brandura, vivacidade e inabalável amor.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and a Mitacs-CALAREO Globalink Research Award (Canada).

I thank my parents, Roberto and Luciana, for their support and trust placed in me throughout the development of this work. I thank my friends, Ana Letícia, Douglas Martins, Douglas Silva, Larissa and Matheus, for their camaraderie presented in the form of countless conversations after (or in the middle of) a day of work, and for their patience when they became heart to hearts.

I thank the Professors Ricardo Custódio, Daniel Panario, Lucia Moura, Jean Martina, Ricardo Dahab and Jintai Ding, who helped improve this research in several ways, mainly through their hospitality in some latitudes (including virtual ones), supervision, availability and critical assessments.

Finally, I thank Poliana, for being part of each and every day that made up this journey, for her tenderness, wit and unfaltering love.

“Anyone, from the most clueless amateur to the best cryptographer, can create
an algorithm that he himself can’t break.” (SCHNEIER, 1998)

RESUMO

O esforço computacional necessário para enfraquecer algoritmos de criptografia assimétrica atualmente usados é proveniente de problemas da teoria de números e álgebra. Com o advento de computadores quânticos, tais problemas tornam-se passíveis de resolução em tempo polinomial. A área de pesquisa formada por criptossistemas assimétricos independentes de tais problemas é chamada de criptografia pós-quântica. Um exemplo de problema onde é desconhecido se sua resolução é acelerada por computadores quânticos é o problema de resolução de sistemas de equações, alicerce para a criptografia baseada em polinômios multivariados. Chaves públicas e privadas desses algoritmos são representadas por sistemas de equações proibitivamente extensos, mas operações com mensagens são demasiado eficientes. Este comportamento é facilmente identificável no esquema de assinatura digital Rainbow, atualmente finalista no processo de padronização de criptografia pós-quântica liderado pelo Instituto Nacional de Padrões e Tecnologia dos Estados Unidos. O esquema Rainbow é baseado em polinômios de "óleo e vinagre", que têm resistido a métodos criptanalíticos até o presente momento. Neste trabalho, um método para reduzir as chaves privadas do esquema Rainbow é proposto, baseado na substituição antecipada de valores em polinômios de chaves privadas, permitindo representações mais compactas. Uma consequência imediata deste fato é que assinaturas são geradas com um nível reduzido de aleatoriedade, mas são apresentados argumentos que mostram a insignificância estatística deste fato em comparações com assinaturas convencionais. O método proposto leva a redução de chaves privadas em até 6.5 vezes, e pode ser combinado com estratégias que reduzem chaves públicas. Entretanto, é exposto que tal método leva a um ataque de recuperação de chaves eficiente, que invalida totalmente a segurança do esquema de assinatura digital derivado. Este trabalho analisa como o ataque é implementado na prática, e demonstra que uma generalização do método proposto é ainda insegura para vários parâmetros do esquema.

Palavras-chave: Assinatura digital. Criptografia pós-quântica. Criptografia baseada em polinômios multivariados. Rainbow.

RESUMO EXPANDIDO

Introdução

Segurança nas comunicações entre dispositivos eletrônicos é um requisito frequentemente desejado, em vista do acréscimo exponencial de troca digital de informações a cada dia. Neste contexto, a criptografia de chaves públicas fornece arcabouços matemáticos para diferentes casos de uso. Particularmente, esquemas de assinatura digital são utilizados para prover autenticidade, integridade e não-repúdio de mensagens e seus remetentes. A segurança de assinaturas digitais utilizadas atualmente é baseada em problemas matemáticos cuja resolução é computacionalmente impraticável até o momento, como a fatoração de inteiros e o logaritmo discreto. Entretanto, em modelos computacionais recentes, como na computação quântica, são propostos algoritmos que podem resolver tais problemas matemáticos mais rapidamente. Desse modo, observa-se a necessidade de analisar algoritmos criptográficos baseados em problemas matemáticos que não são afetados por computadores quânticos. Esta disciplina chama-se criptografia pós-quântica, e existe um esforço considerável de pesquisa na criação de tais métodos, que adicionalmente podem ser executados em computadores convencionais, com o intuito de substituir algoritmos utilizados atualmente. Em especial, os criptossistemas baseados em polinômios de várias variáveis são extremamente eficientes na criação e verificação de assinaturas, porém suas chaves têm tamanho proibitivo em comparação com métodos utilizados atualmente. Assim, demonstra-se uma necessidade da pesquisa em estratégias para redução de chaves em esquemas de assinatura digital baseados em polinômios multivariados.

Objetivos

O objetivo principal deste trabalho é a discussão de estratégias para redução de chaves no esquema de assinatura digital Rainbow. Mais precisamente, são abordadas consequências de tais estratégias em relação à segurança dos esquemas derivados propostos, levando em conta métodos de criptanálise algébrica existentes na literatura. Neste contexto, deseja-se identificar estratégias inseguras apresentadas na literatura e isolar suas características comuns, a fim de prover recomendações na construção de esquemas de assinatura digital derivados do esquema Rainbow com chaves reduzidas. Consequentemente, o objetivo específico central é a proposição de uma estratégia original para redução de chaves através de uma modificação no esquema Rainbow, dissimilar dos métodos apresentados na literatura. Após, discute-se os efeitos da modificação no âmbito do desempenho e da segurança do esquema de assinatura digital resultante.

Metodologia

Os objetivos propostos são concretizados através de uma revisão sistemática da literatura, iniciada com uma pergunta de pesquisa relacionada à segurança de estratégias de redução de chaves em esquemas de assinatura digital derivados do esquema Rainbow. Para tanto, uma revisão bibliográfica exaustiva é realizada, identificando métodos de redução de chaves similares, e criptanálise relacionada à inspeção da segurança de tais estratégias. A classificação é feita de acordo com o tipo de benefício obtido, ou seja, se apenas chaves privadas, públicas, ou ambas são reduzidas. Em seguida, procede-se ao estudo da nova estratégia de redução de chaves aplicada ao esquema Rainbow, distinta das encontradas. A aferição da segurança do esquema resultante é feita através de métodos estatísticos e algébricos obtidos da revisão bibliográfica, observando os artefatos gerados pelo criptossistema, como chaves e assinaturas. Por fim, tamanhos de chave são comparados entre o esquema modificado e original a fim de observar quantitativamente o efeito da alteração.

Resultados e discussão

A revisão bibliográfica realizada demonstra que a modificação do esquema Rainbow, a fim de obter reduções nos tamanhos de suas chaves, frequentemente facilita a aplicação de métodos algébricos que reduzem drasticamente a segurança do esquema resultante. Desse modo, uma nova estratégia é proposta a fim de reduzir chaves privadas do esquema Rainbow, baseada em um método não explorado na literatura, e compatível com outras modificações do esquema subjacente. A estratégia é baseada na redução da aleatoridade de assinaturas através de uma substituição de variáveis prematura na chave privada. A análise estatística de assinaturas e chaves mostra que estes artefatos são indistinguíveis de outros produzidos sem a aplicação do novo método. Chaves privadas são reduzidas em até sete vezes para certos parâmetros, e em conjunto com métodos que reduzem chaves públicas, pares de chaves do esquema resultante são até quatro vezes menores, em comparação com o esquema Rainbow sem modificações. Entretanto, a análise algébrica de assinaturas mostra que a modificação proposta difunde informações da chave privada em cada assinatura. Desse modo, a reunião de várias assinaturas produzidas pelo esquema de assinatura digital modificado permite o cálculo de uma chave privada equivalente de maneira eficiente.

Considerações finais

Modificações feitas ao esquema Rainbow com o intuito de reduzir chaves privadas e públicas tornam o esquema resultante extremamente suscetível a ataques algébricos que reduzem sua segurança. Isto é demonstrado através de uma revisão sistemática da literatura, onde observou-se que apenas uma quantidade limitada de estratégias são consideradas seguras para a redução de chaves do esquema Rainbow. Ainda, um novo método com este intuito foi proposto. A estratégia é superficialmente resistente a métodos estatísticos e algébricos. Entretanto, um método para recuperar uma chave privada equivalente foi proposto a fim de desprovar a segurança do esquema de assinatura digital subjacente. Isto habilita uma entidade maliciosa a forjar assinaturas com baixo esforço computacional. Portanto, consolida-se que modificações ao esquema Rainbow a fim de reduzir chaves são tênues. Deste modo, sugere-se que estratégias sejam aplicadas ao esquema a fim de prevenir a redução de aleatoridade na geração de chaves e assinaturas.

Palavras-chave: Assinatura digital. Criptografia pós-quântica. Criptografia baseada em polinômios multivariados. Rainbow.

ABSTRACT

The computational difficulty of breaking public-key cryptographic algorithms in current use is based on problems from number theory and algebra. With the advent of quantum computers, it becomes feasible to solve such problems in polynomial time. The discipline formed by public-key cryptosystems independent of such problems is called post-quantum cryptography. One example of a problem that is not known to be solved more efficiently by quantum algorithms is the polynomial system solving problem, which is the basis for multivariate cryptography. Prohibitively large polynomial systems represent private and public keys of these algorithms, but operations on messages are very efficient. The Rainbow multivariate signature scheme, currently a finalist of the NIST Post-Quantum Cryptography Standardization Process, is no exception to this norm. Rainbow is based on Oil–Vinegar polynomials, which have resisted known cryptanalysis to this day. We propose a method to reduce Rainbow private keys based on the early substitution of values on the polynomials of private keys, allowing for their compact representation. A direct consequence of this fact is that signatures are generated with a smaller level of randomness, but we argue that this is not statistically discernible from conventional Rainbow signatures. Our method reduces private keys by up to a factor of 6.5 and can be combined with strategies that reduce public keys. However, we find out that our strategy leads to an efficient key recovery attack, which completely undermines the security of the scheme. By analyzing how the attack proceeds in practice, we further argue that a generalization of our previous method is still insecure for several parameter sets.

Keywords: Digital signatures. Post-quantum cryptography. Multivariate cryptography. Rainbow.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Cryptography is the discipline in which techniques for achieving secure communications are studied. Historically, information was hidden from adversaries by scrambling its contents with an algorithm that received a single key as input. The possession of the key is paramount in obtaining the original information. A well-designed algorithm would prevent its discovery by any means other than the usage of the specific key. Said key can *encrypt* plain text or *decrypt* the corresponding ciphered text. This practice, known as symmetric cryptography, persists nowadays with increasingly sophisticated algorithms.

Storage, transportation, and communication of such keys are sensitive procedures. More precisely, an insecure channel can yield a cryptographic key due to a logistical failure or a malicious actor that controls part of the channel. With the advent of computers and the consequent facilitation of communications, this situation became apparent to a greater extent. In order to solve this problem, a new paradigm of cryptography was introduced in (DIFFIE; HELLMAN, 1976), called asymmetric, or *public-key cryptography*. Within this system, an entity holds a key pair, composed of a private key and a public key.

Key pairs are mathematically bound to a computationally hard problem to solve, which is the basis for the security of the resulting cryptosystem. It should be computationally hard to recover a private key from its corresponding public key. There exist various methods of constructing a public-key cryptosystem, with direct consequences on its resulting security, performance, and assurance of distinct properties. Notably, public-key cryptography provides a digital alternative to handwritten signatures, which are one of the ubiquitous ways to ensure trust when a contract is sealed between parties.

Signatures in the form of seals, stamps, or ink drawings present various logistical and security shortcomings. For instance, individuals are advised to be physically present to sign any documents, for they cannot be truthfully identified otherwise. Additionally, this kind of personal calligraphy or symbolism can be forged with little effort. A modern solution is found within mathematical frameworks known as *signature schemes*, that enable explicit assertions on the security of signatures.

A vibrant discipline of public-key cryptography, signature schemes are often associated with digital systems, in which transit of sensitive messages is again expected to be secure. The initial public discovery of digital signature schemes is given in (RIVEST; SHAMIR; ADLEMAN, 1978) and still widely used at the time of the writing. The RSA signature scheme is based on the difficulty of integer factorization. Other schemes, such as Ed25519 (BERNSTEIN et al., 2012), are based on the difficulty of calculating a discrete logarithm. There is no known polynomial-time algorithm to solve such problems on a classical computer.

Classical computers are electronic, using circuits to perform computations. However, quantum computers have emerged as a new paradigm, in which computations are performed outside the scope of classical mechanics. Although these computers are physically hard to implement, algorithms that use quantum phenomena have already been proposed and present a provable speed-up compared to classical computers. One of these algorithms (SHOR, 1997) simplifies the complexity of factoring integers and calculating discrete logarithms. Consequently, it can be theoretically used to break widely used signature schemes.

Post-quantum cryptography addresses this issue by defining public-key cryptosystems that are either not known to be affected by quantum computers, or provably so (BERNSTEIN; BUCHMANN; DAHMEN, 2008). Such algorithms operate on classical computers, and as such, post-quantum cryptography does not encompass quantum algorithms for cryptography. While a new discipline, post-quantum cryptography has generated considerable research effort to replace current cryptosystems with comparatively practical, quantum-safe counterparts. A noteworthy example is the standardization process by the National Institute of Standards and Technology (NIST) (ALAGIC et al., 2019; ALAGIC et al., 2020), which houses proposals based on several distinct mathematical structures as the theoretical bases for underlying problems.

1.1 SCOPE OF OUR RESEARCH

Solving polynomial systems over many variables in the quantum setting is a computationally hard problem, as we discuss in Section 2.2. The discipline of cryptographic primitives based on this problem is known as *multivariate public-key cryptography* or solely multivariate cryptography. Families of multivariate cryptosystems are defined according to how finite fields are employed in the computations, for instance, using a base field and an extension field. Specific improvements are then obtained, such as trade-offs between key and signature sizes, or larger security parameters.

Multivariate cryptosystems are extremely time- and space-efficient when operating over messages. However, public and private keys are defined by large polynomial systems with little opportunity for compression or sparseness. Large key sizes frequently impose constraints on the use of signature schemes in devices of limited storage and processing power. In multivariate cryptography, keys are often orders of magnitude larger than those used in conventional signature schemes.

In this thesis, we solely discuss multivariate signature schemes based on the Oil–Vinegar principle. More precisely, Unbalanced Oil and Vinegar (UOV) (KIPNIS; PATARIN; GOUBIN, 1999) and its generalization Rainbow (DING; SCHMIDT, 2005) receive primary attention. Such schemes perform computations over a single field and present a balanced nature with simple, elegant definitions. Signature schemes of a different structure such as HFE (PATARIN, 1996), or multivariate public-key encryption schemes such as SRP (DUONG; PETZOLDT; TAKAGI, 2016), are not addressed. Furthermore, we specifically discuss strategies for reducing key sizes of Rainbow-like signature schemes and the security implications of such methods.

1.2 CONTRIBUTIONS AND OUTLINE OF THIS THESIS

We intend to provide the reader with a complete overview of the mathematical background necessary to understand the family of Oil–Vinegar signature schemes and related discussions of its security. Thus, in Chapter 2, we start by introducing the mathematical concepts used in the description of such schemes and then proceed to present the UOV and Rainbow signature schemes.

We develop upon Rainbow on Chapter 3, where we propose a novel approach to reduce the private key size with three possible implementations. We discuss the consequences of our modifications via statistical analysis and show its consequences on key pair sizes for several parameter sets. The contents of this chapter were published as a peer-reviewed paper in a conference proceeding (ZAMBONIN; BITTENCOURT; CUSTÓDIO, 2019).

In Chapter 4, we show that our modification is susceptible to a key recovery attack and discuss the practical efforts associated with applying the attack. Moreover, we argue that the general form of our proposal, even in its simplest form, considerably reduces the security levels of several Rainbow parameter sets.

Finally, in Chapter 5, we summarize and finish our discussion with pointers to open research topics.

1.3 RELATED WORK

In this section, we use the previously well-defined boundaries with regards to the scope of our work to perform the necessary search in the literature. We recall that our main object of study consists of digital signature schemes which make use of Oil–Vinegar polynomials in their construction. More precisely, it is known that key sizes for instances of such schemes are currently impractical. Thus, we inspect works that aim to mitigate this situation to identify common strategies and pitfalls.

We conduct our literature review by using the Google Scholar meta-indexer tool, which covers most scientific literature providers, and allows the user to delimit the desired scope efficiently. We identify three critical works in the history of signatures based on Oil–Vinegar polynomials (PATARIN, 1997; KIPNIS; PATARIN; GOUBIN, 1999; DING; SCHMIDT, 2005) and search for works that directly cited at least one of those. We consider only publications written in English that additionally are not patents. From this set, we select relevant papers and present a summary below.

We note that Rainbow is in the third round of the NIST standardization process as a finalist (ALAGIC et al., 2020, Sec. 3.20). Throughout this work, we discuss the algorithm as submitted to the second round (DING et al., 2019). Currently, there is no available third-round submission for inspection. However, NIST strongly suggests that no significant changes are made to the algorithm, and thus we expect that our conclusions also hold for the newest submission.

In this section, we mention by name several methods that have thwarted schemes. Such techniques of algebraic cryptanalysis are discussed in Section 3.1.4. In particular, equivalent keys can also be used to reduce private key sizes. A summary of the theory is given in Section 2.2, and in Section 2.3 within the context of UOV and Rainbow.

1.3.1 Improvements to private key size

Rainbow is the most popular multivariate signature scheme based on Oil–Vinegar polynomials. However, schemes of that class that feature modifications to reduce key sizes have been proposed even before Rainbow itself was proposed. Chronologically, the first schemes with this feature are tame transformation schemes (TTS) (CHEN; YANG; PENG, 2002), in which equations of the private key are required to present a minimum level of sparseness. This strategy was attacked with various algebraic techniques, e.g. MinRank and the UOV attack, and corrected several times, as summarized in (DING; SCHMIDT; YIN, 2006).

A similar idea introduced at roughly the same time is the use of stepwise triangular systems (STS) in their generalized form (WOLF; BRAEKEN; PRENEEL, 2006). The shape of their private keys is derived from restrictions on polynomials of the central map. This strategy indirectly enables a reduction in private key size. STS were found to be insecure in the same work that designed the general classification. The authors propose an inversion attack that recovers the original message from its corresponding signature, and they additionally show that it is possible to build an equivalent private key.

STS and TTS were later found to be particular cases of the general Rainbow construction (DING et al., 2008), due to their similarities in the usage of layers and the central map inversion procedure. These advances, as well as further cryptanalysis of Rainbow itself (BILLET; GILBERT, 2006) with the MinRank attack, led to the proposal of new parameters for TTS and Rainbow (DING et al., 2008). This improvement was achieved with "Rainbow-Band-Separation" (RBS), an extension of the UOV "reconciliation" attack.

Further modifications were proposed as an attempt to salvage previous schemes (TSUJII et al., 2010), but every such variant was broken through a key recovery attack (THOMAE; WOLF, 2012) which stems from the generalization of the theory of equivalent keys. A consequence of this theory is that the associated matrices of the affine transformations that hide the central map structure may be thoroughly simplified. It is explored in (CZYPEK; HEYSE; THOMAE, 2012), for instance, to insert instances of UOV and Rainbow into low-resource devices.

We note that in the case of STS and TTS, reducing the private key was not the primary intention of the authors, and it is thus not trivial to estimate such gains through direct comparison to Rainbow. Indeed, the reduction of keys is a known strategy to increase the performance of operations on signatures. On the other hand, several proposals in the literature focus on the modification of Rainbow with the sole purpose of reducing key sizes. This motivation partially stems from recent efforts to present quantum-safe cryptography as a valid alternative to traditional digital signature schemes (BERNSTEIN; BUCHMANN; DAHMEN, 2008).

A variant of Rainbow based on quadratic forms (YASUDA; TAKAGI; SAKURAI, 2013) features a trade-off between the private key and public key sizes to increase the performance of the signature generation step. It replaces the central map with a small invertible matrix, reducing the private key size by up to 91.2% and increasing the public key threefold. However, it was found to be insecure afterward (HASHIMOTO, 2016) by algebraic cryptanalysis leading up to the discovery of equivalent keys.

NC-Rainbow (YASUDA; SAKURAI; TAKAGI, 2012) is proposed as a novel strategy, based in non-commutative rings, used to reduce a private key by up to 75%. However, it was shown by independent researchers to be insecure through a combination of rank attacks and the UOV attack (THOMAE, 2012; HASHIMOTO, 2015). The variants MB-Rainbow (YASUDA et al., 2013) and NT-Rainbow (YASUDA; TAKAGI; SAKURAI, 2014b) are based on the insertion of, respectively, sparse diagonal and non-triangular matrices into the central map, to reduce the private key by up to 40%. The authors merged MB- and NC-Rainbow into a single scheme MNT-Rainbow (YASUDA; TAKAGI; SAKURAI, 2014a), shortening private keys by up to 76%.

The parameters of MB- and NT-Rainbow were deemed vulnerable to the RBS attack by the authors of (PENG; TANG, 2017). Circulant Rainbow is also proposed in that work, based on the concept of rotating relations introduced to the central map. It gives a reduction of the private key by up to 45%. This approach and its analogous UOV variant (PENG; TANG, 2018) were broken shortly after with an application of the UOV attack (HASHIMOTO, 2019). The strategy of MB-Rainbow was applied to UOV in (TAN; TANG, 2015), alongside with the replacement of certain parts of the private key by values generated through linear recurring sequences. While these strategies reduce the private key by up to 89.1%, they were found to be insecure (PARK, 2018) with the discovery of an attack to find equivalent keys efficiently.

In the ELSA variant (SHIM; PARK; KOO, 2017), a hidden layer of quadratic equations is inserted to simplify the private key and obtain a size reduction of up to 88%. Nonetheless, it was found that such structure introduced shortcuts enabling an attacker to find equivalent keys (HASHIMOTO; IKEMATSU; TAKAGI, 2019). The SOV signature scheme (SHIM; PARK; KOO, 2020) is built by preventing overlap of quadratic terms in the central map and assigning a specific rank to the matrices associated with its polynomials, reducing the private key size by up to 91.9%.

Some non-intrusive approaches are also found in the literature. The authors of Lite-Rainbow-0 (SHIM; PARK; BAEK, 2015) propose the replacement of the entire private key with a small pseudorandom number generator (PRNG) seed. The private key is reduced by a factor of approximately 99.8%, but the cost for signature generation increases in roughly 70%. In (BORGES; PETZOLDT; PORTUGAL, 2012), it is suggested to use the RC4 stream cipher in the private key generation of UOV, presenting analogous decreases in the private key size but no noticeable effects on signature generation performance. Such an approach also works with Rainbow (DORNELLES; LARA; HENRIQUES, 2019). A similar result is also given in another work (SEO et al., 2014), in which AES-CTR is employed as the PRNG of choice.

The pre-computation of UOV signatures such that the private key is not required is proposed in (CHEN et al., 2016). This "online-offline" approach does not modify the key itself but can be used to yield signatures without the need to load the private key, thus reducing total storage requirements. Finally, our proposal (ZAMBONIN; BITTENCOURT; CUSTÓDIO, 2019) is discussed in more detail in (BITTENCOURT, 2019) and further expanded in Chapter 3.

1.3.2 Improvements to public key size

In the case of modifications to the public key, there are not as many approaches as for private keys. The first strategy is based on field lifting, where the resulting LUOV scheme (BEULLENS; PRENEEL, 2017) has the central map and public key lifted to an extension field, such that coefficients of those polynomial systems are now smaller. It is a trade-off between public key size and signature size, but general enough to be compatible with other public-key reduction strategies. LUOV has been submitted to the standardization process by NIST (ALAGIC et al., 2019), and it was subsequently attacked with a new strategy based on differentials between the base and extension fields (DING et al., 2020). Thus, it was not considered to the third round of the process (ALAGIC et al., 2020, Sec. 3.24). An extension of the LUOV proposal to Rainbow (DUONG; LUYEN; TRAN, 2020) was submitted before any attacks were known, but it is unclear if it is affected by the differential method.

A development by the name of BACUOV (SZEPIENIEC; PRENEEL, 2019) forces matrices representing the central map and public key to be block-anti-circulant. It allows for a compression of both keys in the key pair, although only public key size improvements are featured on the work mentioned above. Still, the authors of (FURUE et al., 2020) show that the security levels of BACUOV are smaller than initially claimed by manipulating the public key and applying the UOV attack.

We also classify the work in (SZEPIENIEC; BEULLENS; PRENEEL, 2017) as relevant since it is a general method motivated by the context of public key infrastructures, in which the size of a public key and signature bundle must be minimized. The strategy, which consists of a trade-off between signature and public key sizes with their combined length still reduced, was initially proposed for multivariate trapdoors, and subsequently generalized to work in different security models and other signature schemes (BEULLENS; PRENEEL; SZEPIENIEC, 2018).

The approach proposed in (PETZOLDT; BULYGIN; BUCHMANN, 2010a) is, to the best of our knowledge, the most scrutinized method for public-key reduction without compromises to the signature size. It was initially used to create cyclic UOV, which has cyclic relations built into the public key, reducing its size by up to 80.5%. It explores the fact that specific parts of the public key do not contribute to the security of the scheme and can be structured at will. A detailed discussion of the mathematical relations that allow for these improvements and their limitations are shown in Subsection 3.1.2. Schemes that make use of this framework are discussed below.

A strategy analogous to a manipulation between base and extension fields is given in (PETZOLDT et al., 2011). The resulting scheme is 0/1 UOV, in which coefficients of the public key are in $\mathbb{F}_2$. The proposal consists of creating the matrix associated with the structured part of the key according to a graph structure. Such a graph describes a relationship of coefficients that, while sparse, does not facilitate known attacks over UOV, reducing the public key size by up to 88.6%. However, it is unknown if this proposal can be extended to Rainbow.

The CyclicRainbow variant (PETZOLDT; BULYGIN; BUCHMANN, 2010b) is a direct extension of cyclic UOV. It is featured on the Rainbow submission for the second round of the standardization process conducted by NIST (DING et al., 2019). It reduces public keys by up to 53.8%, and the associated key generation algorithm is comparably efficient to Rainbow (PETZOLDT, 2020). Another strategy is the usage of linear recurring sequences to generate such unimportant parts of the public key. It is summarized in two works (PETZOLDT; BULYGIN; BUCHMANN, 2011; PETZOLDT; BULYGIN, 2012) and reduces public keys by up to 56.6%.

1.3.3 Improvements to total key pair size

Additionally, there are works in the literature that manage to reduce both private and public key sizes through novel strategies. The authors of LWRS (ZHANG et al., 2012) create a scheme with low-resource devices in mind. The left affine transformation of the usual bipolar construction is dropped, and thus the private key is reduced accordingly. The “minus” method (WOLF, 2005, Sec. 3.2.1), which consists of removing several equations in order to reduce storage requirements, is applied to the public key.

A modification that consists of adding cross-terms of oil variables in the last layer to the last two layers of Rainbow is proposed in (TAN et al., 2016). With this method, which is also applicable to UOV, the authors reduce Rainbow private and public keys by up to 46.2% and 36.8%, respectively. The matrices associated with the public and private keys of the Hufu-UOV scheme (TAO, 2019) have special Toeplitz and circulant constructions that, for instance, enable compression of the public key by up to a factor of 4.5.

The variant cubic UOV (NIE et al., 2015) introduces several cubic polynomials in the central map of the private key. A consequence is that both keys of this variant are smaller compared to the original UOV scheme, although the focus of the original work is on reducing public keys. However, it was found to be insecure (HASHIMOTO, 2017) through the recovery of equivalent keys.

Repaired versions named CSSv and SVSv were proposed (DUONG et al., 2016). The former removes most cubic polynomials and further unnecessary structure from the key pair, while the latter does not feature cubic polynomials at all and uses a sparse central map. These proposals achieve further decrease in key sizes, being comparable to Rainbow in some instances. However, independent researchers found both strategies to be flawed (SHIM; KOO;

PARK, 2017; HASHIMOTO, 2017) to HighRank and key recovery attacks using equivalent keys.

There have been commendable efforts in the creation, validation, and correction of signature schemes based on Oil–Vinegar polynomials. However, most works that introduce any kind of structure into the key pair eventually succumb to algebraic cryptanalysis. We perform this review to present evidence that such approaches should be applied with caution and to suggest a distinct approach, as presented in Chapters 3 and 4.

2 MATHEMATICAL BACKGROUND

In this chapter, we give the mathematical toolbox necessary to comprehend the remainder of this work adequately. In Section 2.1, we define basic algebraic structures that rule the operations of mathematical structures within our scope. In Section 2.2, a brief panorama of concepts associated with multivariate cryptography is given. In Section 2.3, we discuss the Unbalanced Oil and Vinegar signature scheme, its generalized form Rainbow, and how cryptanalytic methods impact this trapdoor.

We use the following symbols throughout this work. The symbol $\overset{r}{\leftarrow}$ is read as "chosen randomly from". The symbol $\approx_\varepsilon$ means that two numbers are equal within a precision of ε. The usual function composition is given by the symbol $\circ$, and the inverse of a function f is given by f^{-1}. The usual mean and standard deviation functions for a set of elements S are respectively given by $\mu(S)$ and $\sigma(S)$. The concatenation of two elements up to context is given by $||$.

2.1 ALGEBRAIC STRUCTURES

We define below the elementary algebraic structures known as groups, rings, fields, and vector spaces. We make this clear distinction since there are several underlying structures used in multivariate cryptography that, for instance, operate in the context of a finite field but form only a group or ring. Most definitions are taken from (DUMMIT; FOOTE, 2003) and (MULLEN; PANARIO, 2013), and we refer the reader to these references for proofs of the facts given below.

Definition 2.1.1. Given a set G and any $g_1, g_2, g_3 \in G$, a *binary operation* $\star$ *on a set* G is a function $\star : G \times G \to G$, where we denote $\star(g_1, g_2)$ as $g_1 \star g_2$. A binary operation $\star$ on a set G may also satisfy the following conditions, for all $g_1, g_2, g_3 \in G$:

- it is *associative* if $g_1 \star (g_2 \star g_3) = (g_1 \star g_2) \star g_3$;
- it is *commutative* if $g_1 \star g_2 = g_2 \star g_1$;
- it is *closed under G* if $g_1 \star g_2 \in G$.

Definition 2.1.2. Given a set G, any $g \in G$ and a binary operation $\star$ on G, a *group* is an ordered pair $(G, \star)$ that satisfies the following axioms:

1. The binary operation $\star$ is *closed under G*.
2. The binary operation $\star$ on G is *associative*.
3. There exists a unique element $e \in G$, *the identity of* G, such that $g \star e = e \star g = g$.
4. There exists a uniquely determined element $h \in G$, *the inverse of* g, such that $g \star h = h \star g = e$, and indeed the inverse of h is g.

We are particularly interested in specific types of groups and transformations between those. Thus, we also give the following definitions, which are useful throughout the work.

Definition 2.1.3. Given a group $(G,\star)$, if G is a finite set, then the group is *finite*. The number of elements in a group is its *order*. A group is never "empty", due to the existence of the identity element.

Definition 2.1.4. Given a group $(G,\star)$, if the binary operation $\star$ on G is commutative, then the group is commutative, or *abelian*.

If clear from the context, we hereafter refer to group-like structures by the name of its base set with a sans-serif font and omit the associated binary operations. By restricting the binary operation to specific elements of the underlying set, we obtain the intuitive structure defined below.

Definition 2.1.5. Given a group $\mathsf{G} = (G,\star)$ and $H \subseteq G$, if $\mathsf{H} = (H,\star)$ forms a group, then H is a *subgroup* of G.

Furthermore, often it is convenient to interpret elements as those of a different group. However, the binary operation has to be "preserved" throughout this mapping. We precisely define this below.

Definition 2.1.6. Given two groups $\mathsf{G} = (G,\star), \mathsf{H} = (H,\diamond)$ and any $g_1, g_2 \in G$, a function $\varphi : \mathsf{G} \to \mathsf{H}$ is a *homomorphism* if $\varphi(g_1 \star g_2) = \varphi(g_1) \diamond \varphi(g_2)$.

Definition 2.1.7. Given two groups G and H, if a homomorphism $\varphi : \mathsf{G} \to \mathsf{H}$ is a bijection, it is an *isomorphism*. If $\mathsf{G} = \mathsf{H}$, φ is an *endomorphism*. If the function is simultaneously an isomorphism and an endomorphism, it is an *automorphism*.

Proposition 2.1.8. *Given a group* G *and the set of all its automorphisms* $\mathsf{Aut}_{\mathsf{G}}$*, then* $\mathsf{Aut}(\mathsf{G}) = (\mathsf{Aut}_{\mathsf{G}}, \circ)$ *is a group.*

Groups carrying usual addition or multiplication operations are *additive* and *multiplicative* groups, respectively. Common examples are the additive group under $\mathbb{R}$, that is, $(\mathbb{R},+)$, and the multiplicative group of integers modulo n, $(\mathbb{Z}/n\mathbb{Z})^{\times}$. Adding further operations to groups leads to more usual arithmetic structures.

Definition 2.1.9. Given a set R, any $r_1, r_2, r_3 \in R$, and two binary operations $+$ and $*$ on R, a *ring* is an ordered 3-tuple $(R,+,*)$ satisfying the following axioms:

1. The ordered pair $(R,+)$ forms an *abelian group*.

2. The binary operation $*$ on R is *associative*.

3. The binary operation $*$ *left- and right-distributes over* $+$, that is, $r_1 * (r_2 + r_3) = (r_1 * r_2) + (r_1 * r_3)$ and $(r_2 + r_3) * r_1 = (r_2 * r_1) + (r_3 * r_1)$.

Definition 2.1.10. Given a ring $(R,+,*)$, if the binary operation $*$ is commutative, then the ring is *commutative*.

We hereafter refer to 0 as the *additive identity* and to 1 as the *multiplicative identity* of a ring, if it exists. Analogously, for any element r of a ring, we denote $-r$ as the *additive inverse of* r, and r^{-1} as the *multiplicative inverse of* r, if it exists. Indeed, these inverse binary operations are respectively known as *subtraction* and *division*.

Definition 2.1.11. Given a ring $(R,+,*)$, the smallest integer n such that $1+1+\cdots+1=0$, where 1 is added $n-1$ times, is the *characteristic* of the ring. If such n does not exist, then the characteristic is zero.

Definition 2.1.12. Given a ring R and any $r \in R, r \neq 0$, if there exists a unique element $1 \in R$ such that $r*1 = 1*r = r$, then R is a *ring with unity*.

Definition 2.1.13. Given a ring R with unity, if $(R \setminus \{0\}, *)$ forms a group, then R is a *division ring*.

The ring of integers equipped with the usual addition and multiplication operations $(\mathbb{Z},+,*)$ is a commutative ring with unity. The quaternions, a number system employed, for example, in three-dimensional computer graphics, form a non-commutative division ring with its usual binary operations.

Definition 2.1.14. A commutative division ring is a *field*.

Particularly, Galois fields or *finite fields* are at the heart of various branches of mathematics and computer science, and especially cryptography. They are denoted as $\mathbb{F}_q$, $q = p^n$ where the prime number p is the *characteristic* of the field, and q is omitted for brevity if appropriate.

Proposition 2.1.15 ((MULLEN; PANARIO, 2013, Cor. 2.18))**.** *The order of a finite field is always of the form p^n with p prime and $n \in \mathbb{N}$. If $n = 1$, then it is a* prime field.

For instance, the prime field $\mathbb{F}_p$ is composed of the set $\{0,\ldots,p-1\}$ equipped with the usual addition and multiplication operations modulo p. The respective inverse operations are subtraction modulo p and the modular multiplicative inverse, which is calculated by solving Bézout's identity with the extended Euclidean algorithm. Apart from the examples above, we can also construct more elaborate structures over mathematical objects that live outside the usual sets of numbers. Specifically, we aim to define non-prime fields with the following framework.

Definition 2.1.16. Given $k \in \mathbb{N}_0$, a set of *coefficients* A with $a_0,\ldots,a_k \in A$, and an *indeterminate* x, a *univariate polynomial* is an expression of the form

$$\sum_{i=0}^{k} a_i x^i = a_k \cdot x^k + \cdots + a_2 \cdot x^2 + a_1 \cdot x + a_0. \tag{2.1}$$

Each of the terms in the summation is a *monomial*.

Definition 2.1.17. Given $n \in \mathbb{N}$, a polynomial over a sequence of indeterminate $\mathbf{x} = (x_1, \ldots, x_n)$ is a *multivariate polynomial*. The highest sum of the exponents of the indeterminates in a monomial with a non-zero coefficient is the *degree* of the polynomial, or $\deg(\cdot)$. If all monomials have the same degree, then the polynomial is *homogeneous*.

Polynomials of degree zero, one, two and three are respectively called *constant*, *linear*, *quadratic* and *cubic*. We note that addition and subtraction between polynomials is done by simply operating coefficients of monomials with the same product of variables, that is,

$$\sum_{i=0}^{k} a_i \mathbf{x}^i \pm \sum_{i=0}^{k} b_i \mathbf{x}^i = \sum_{i=0}^{k} (a_i \pm b_i) \mathbf{x}^i \tag{2.2}$$

where $\mathbf{x}^i = x_1^{i_1} \cdots x_n^{i_n}$ and $i_1, \ldots, i_n \in \mathbb{N}_0$. Multiplication is calculated through repeated applications of the distributive law. Division is calculated with the Euclidean algorithm, considering $x^0 < x^1 < \cdots < x^k$ as an intuitive ordering to identify divisors and dividends, in the case of univariate polynomials. This allows us to define the following structure.

Definition 2.1.18. Given $k \in \mathbb{N}_0$, coefficients $a_0, \ldots, a_k$ in a commutative ring R and indeterminate x, the set of polynomials $\sum_i^k a_i x^i$ form an *univariate polynomial ring* $\mathsf{R}[x]$. Given $n \in \mathbb{N}$ and indeterminates $\mathbf{x}$, the respective set of polynomials form a *multivariate polynomial ring* $\mathsf{R}[x_1, \ldots, x_n]$ or $\mathsf{R}[\mathbf{x}]$.

Definition 2.1.19. A polynomial $f_1 \in \mathsf{R}[\mathbf{x}]$ is *irreducible over* R if $f_1 = f_2 f_3$, with $f_2, f_3 \in \mathsf{R}[\mathbf{x}]$ implies that f_2 or f_3 is in R. Otherwise, f_1 is reducible over R.

We note that an irreducible polynomial over a ring always exists (MULLEN; PANARIO, 2013, Remark 2.1.25). An irreducible polynomial cannot be reduced or divided into simpler polynomials. In the context of multivariate polynomial rings, the division operation is ambiguous if no additional structure is attached. This distinction is discussed in detail in Section 2.2, tailored to the context of multivariate cryptography.

We are now ready to discuss non-prime fields. Let p be a prime number and $n \in \mathbb{N}$. It is well-known that there exists a finite field with p^n elements (MULLEN; PANARIO, 2013, Thm. 2.1.32). Indeed, the finite field $\mathbb{F}_{p^n}$ can be represented as the set of polynomials with coefficients in $\mathbb{F}_p$ and degree smaller than n carrying the usual polynomial arithmetic operations modulo f. Finite fields of the form $\mathbb{F}_{2^n}$ are especially useful in computer science, given that coefficients are restricted to $\{0, 1\}$ and may be represented by bit arrays, with arithmetic done through logic operations.

We remark that a polynomial by itself is simply a mathematical expression, but it is commonly used to define a *polynomial function*. In this case, the indeterminates are the *variables* of the function $f(\mathbf{x})$ associated with the polynomial. The domain and codomain of such a function must be thus well-defined, for instance, by picking a field. Further, the solutions to the associated equation $f(\mathbf{x}) = 0$, if they exist, are the zeros of the function or the *roots* of a polynomial.

Several polynomials may be used to model a complex structure in which they behave concurrently. It is thus useful to have a relevant mathematical concept to represent this situation.

Definition 2.1.20. Given $m \in \mathbb{N}$, a *polynomial system* is a simultaneous sequence of polynomial equations $f^{(1)}(\mathbf{x}) = 0, \dots, f^{(m)}(\mathbf{x}) = 0$, or

$$\begin{cases} f^{(1)}(x_1, \dots, x_n) & = 0, \\ \vdots & \\ f^{(m)}(x_1, \dots, x_n) & = 0. \end{cases} \tag{2.3}$$

A polynomial system may be classified with regards to the relation between m and n. Consider that no equations in the system are linear combinations of any other equations. If this is not the case, then the polynomial system can be manipulated to remove equations or variables, affecting m or n. Then, if $m > n$, the system is *overdetermined*. If $m < n$, the system is *underdetermined*. If $m = n$, the system is *exactly determined*.

Furthermore, a system is *consistent* if there exists at least one solution $x_1, \dots, x_n$ and *inconsistent* otherwise, independently of m and n. Root-finding techniques have been studied extensively, and there exist several different methods to calculate one or all roots (PRESS et al., 2007, Chap. 9). We are mostly focused on solving systems of linear equations, in which all polynomials have $\deg(\cdot) \leq 1$, that can be solved through Gaussian elimination (PRESS et al., 2007, Sec. 2.2).

It is useful to define an algebraic structure that falls outside of the class of ring-like structures to generalize the notion of grouping polynomials into a system.

Definition 2.1.21. Given a finite field $\mathbb{F}$ equipped with binary operations $+$ and $*$, a set V, any elements $s_1, s_2 \in \mathbb{F}$, $\mathbf{v_1}, \mathbf{v_2} \in V$, a binary operation $\boldsymbol{+}$ on V and a binary operation $\times : \mathbb{F} \to V$, a *vector space over* $\mathbb{F}$ is an ordered 3-tuple $(V, \boldsymbol{+}, \times)$ satisfying the following axioms:

1. The ordered pair $(V, \boldsymbol{+})$ forms an *abelian group*.
2. The binary operation $\times$ is *associative*, that is, $s_1 \times (s_2 \times \mathbf{v_1}) = (s_1 * s_2) \times \mathbf{v_1}$.
3. The binary operation $\times$ *left distributes over* $\boldsymbol{+}$, that is, $s_1 \times (\mathbf{v_1} \boldsymbol{+} \mathbf{v_2}) = (s_1 \times \mathbf{v_1}) \boldsymbol{+} (s_1 \times \mathbf{v_2})$.
4. The binary operation $+$ *left distributes over* $\times$, that is, $(s_1 + s_2) \times \mathbf{v_1} = (s_1 \times \mathbf{v_1}) \boldsymbol{+} (s_2 \times \mathbf{v_1})$.

Vector spaces are the object of study of linear algebra. The members of V are *vectors*, and the members of $\mathbb{F}$ are *scalars*. It follows that the binary operation $\times$ is the *scalar multiplication*. If clear from the context, we hereafter name vector spaces by capital boldface letters, without mentioning the associated binary operations. Given $c \in \mathbb{N}$, if members of V are c-tuples of elements of $\mathbb{F}$, the resulting vector space is a *coordinate space* represented by

$\mathbb{F}^c$, with component-wise addition, subtraction and scalar multiplication. We can thus recall Definition 2.1.20 and create a function of several polynomials.

Definition 2.1.22. Given $n,m \in \mathbb{N}$, a polynomial ring $\mathsf{F} = \mathbb{F}[x_1,\ldots,x_n]$ and $f^{(1)},\ldots,f^{(m)} \in \mathsf{F}$, a *polynomial map* is a function $j : \mathbb{F}^n \to \mathbb{F}^m$ defined by an m-tuple of polynomial functions

$$j(x_1,\ldots,x_n) = (f^{(1)}(x_1,\ldots,x_n),\ldots,f^{(m)}(x_1,\ldots,x_n)). \tag{2.4}$$

We give other useful traits of a vector space before returning to polynomial maps.

Definition 2.1.23. Given a vector space $(V,+,\times)$ over $\mathbb{F}$, $\mathbf{v_1},\ldots,\mathbf{v_n} \in V$ and $s_1,\ldots,s_n \in \mathbb{F}$, if the unique solution for the linear combination $s_1 \times \mathbf{v_1} + \cdots + s_n \times \mathbf{v_n} = 0$ is $s_1 = \cdots = s_n = 0$, then the vectors $\mathbf{v_1},\ldots,\mathbf{v_n}$ are *linearly independent*.

Definition 2.1.24. Given a vector space $\mathbf{V}$ and $B = \{\mathbf{v_1},\ldots,\mathbf{v_n}\} \subseteq V$, the set of all linear combinations $S = \{s_1 \times \mathbf{v_1} + \cdots + s_n \times \mathbf{v_n} \mid s_1,\ldots,s_n \in \mathbb{F}\}$ is the *linear span of B*. If the elements of B are linearly independent and $S = V$, then B is a *basis of the vector space* $\mathbf{V}$.

There always exists at least one basis for a vector space (DUMMIT; FOOTE, 2003, p. 409, Prop. 1). Indeed, it is a compact way of representing the entire set of vectors. For example, the *standard basis* is a sequence of unit vectors $(1,0,\ldots,0),\ldots,(0,\ldots,0,1)$ over $\mathbb{F}^n$. The ordered arrangement of the standard basis yields the *identity matrix* $\mathbf{I}_n$. Furthermore, all bases have the same number of elements (DUMMIT; FOOTE, 2003, p. 411, Cor. 4), which is the *dimension* of the vector space, or $\dim(\cdot)$.

Analogously to group- and ring-like structures, there are mappings between vector spaces that must "preserve" both binary operations, akin to Definition 2.1.6. The general form of such functions is presented below.

Definition 2.1.25. Given two vector spaces $\mathbf{V} = (V,+,\times)$ and $\mathbf{V}' = (V',+',\times')$ over the same field $\mathbb{F}$, any vectors $\mathbf{v_1},\mathbf{v_2} \in V$ and any scalar $s \in \mathbb{F}$, a function $\phi : \mathbf{V} \to \mathbf{V}'$ is a *linear transformation* if it satisfies the following conditions:

1. The *addition operation is preserved*, that is, $\phi(\mathbf{v_1}+\mathbf{v_2}) = \phi(\mathbf{v_1}) +' \phi(\mathbf{v_2})$.
2. The *scalar multiplication operation is preserved*, that is, $\phi(s \times \mathbf{v_1}) = s \times' \phi(\mathbf{v_1})$.

We obtain a *vector subspace of* $\mathbf{V}$ by applying the rationale of Definition 2.1.5. Observing how a linear transformation acts on a subspace leads to the following definition.

Definition 2.1.26. Consider a vector space $\mathbf{V}$ over $\mathbb{F}$, a linear transformation $\phi : \mathbf{V} \to \mathbf{V}$, a vector subspace $\mathbf{W}$ of $\mathbf{V}$ and any $\mathbf{w} \in \mathbf{W}$. If $\phi(\mathbf{w}) \in \mathbf{W}$, then $\mathbf{W}$ is an *invariant subspace under ϕ*.

Such functions between coordinate spaces can be represented by polynomial maps. A trivial example of a linear transformation is the identity function $\mathsf{id} : \mathbf{V} \to \mathbf{V}$ such that $\mathsf{id}(\mathbf{v_1}) = \mathbf{v_1}$. Moreover, if $\dim \mathbf{V} = \dim \mathbf{V}'$, there may exist an inverse function $\phi^{-1} : \mathbf{V}' \to \mathbf{V}$ such that

$\phi \circ \phi^{-1} = \mathsf{id}$. We remark that Definition 2.1.7 and Proposition 2.1.8 are analogous in the setting of vector spaces. Indeed, there exists an automorphism group $\mathsf{Aut}(\mathbf{V})$, which is the set of all invertible linear transformations carrying the usual function composition operation. Such group is also known as the *general linear group* $\mathsf{GL}(\mathbf{V})$.

Proposition 2.1.27 ((DUMMIT; FOOTE, 2003, p. 418, Cor. 14))**.** *Consider a vector space* $\mathbf{V}$ *over* $\mathbb{F}$ *with* $\dim(\mathbf{V}) = n$*. Let the* general linear group of degree n, $\mathsf{GL}_n(\mathbb{F})$*, be the set of all invertible matrices of order* $n \times n$ *over* $\mathbb{F}$ *carrying the usual matrix multiplication binary operation. Then, there exists an isomorphism* $\psi : \mathsf{GL}(\mathbf{V}) \to \mathsf{GL}_n(\mathbb{F})$.

We observe that the $\mathsf{GL}_n(\mathbb{F})$ notation does not refer to a vector space. This is due to the remarkable fact that linear transformations may be interpreted as matrices over the base field of the vector space (DUMMIT; FOOTE, 2003, Sec. 11.2). Thus, map compositions and applications can be interpreted as matrix-matrix and matrix-vector multiplications throughout our work. For example, the matrix associated with id is exactly $\mathbf{I}_{\dim \mathbf{V}}$. More generally, given $\phi \in \mathsf{GL}(\mathbf{V})$, its associated matrix $\psi(\phi) = M \in \mathsf{GL}_n(\mathbb{F})$ and a vector $\mathbf{v} \in \mathbf{V}$, then $\phi(\mathbf{v}) = M\mathbf{v}$.

2.2 MULTIVARIATE SIGNATURE SCHEMES

With most mathematical machinery defined, we now introduce the building blocks to create a generic public-key cryptosystem based on multivariate polynomial equations, focusing on digital signature schemes. We first recall two essential components of cryptography, with definitions taken from (GATHEN, 2015, Sec. 7.1) and (GOLDREICH, 2004, Sec. 6.1).

Definition 2.2.1. Given a function $\mathcal{H} : X \to Y$ and any $x \in X$, $\mathcal{H}$ is a *hash function* if calculating $\mathcal{H}(x)$ is deterministic and computationally easy. Usually, a hash function is also compressing, that is, $|x| > |\mathcal{H}(x)|$.

Definition 2.2.2. Given a hash function $\mathcal{H} : X \to Y$ and any $m_1, m_2 \in X$, $\mathcal{H}$ may respect the following properties.

1. *Preimage resistance*: it is computationally unfeasible to find m_1 given $\mathcal{H}(m_1)$.

2. *Second preimage resistance*: it is computationally unfeasible to find m_2 given m_1 such that $m_1 \neq m_2$ and $\mathcal{H}(m_1) = \mathcal{H}(m_2)$.

3. *Collision resistance*: it is computationally unfeasible to find a pair (m_1, m_2) such that $m_1 \neq m_2$ and $\mathcal{H}(m_1) = \mathcal{H}(m_2)$.

A *cryptographic hash function* is a hash function which respects all three properties of Definition 2.2.2. It is the usual method of transforming a variable-length message into a fixed-length *digest*. Usually, the set X is composed of all binary words $\{0,1\}^*$, while Y is $\mathbb{F}_2^k$ for some small $k \in \mathbb{N}$. A well-designed function outputs a random string with no apparent connection to the input, which changes completely even with the smallest correction applied to the original message. It is a handy tool for integrity checking.

Multivariate cryptography is based on public-key cryptography. Let us denote private and public keys as sk and pk, respectively. We give the formal definition of a signature scheme below.

Definition 2.2.3. Given $n \in \mathbb{N}$ and a message $w \in \{0,1\}^*$, a 3-tuple of probabilistic polynomial-time algorithms $(\textsc{Gen}, \textsc{Sig}, \textsc{Ver})$ is a *digital signature scheme* if every signature generated by $\textsc{Sig}$ verifies under $\textsc{Ver}$ for every possible key pair generated by $\textsc{Gen}$. Equivalently,

$$\forall(\mathsf{sk},\mathsf{pk}) \leftarrow \textsc{Gen}(1^n), \quad \Pr[\textsc{Ver}(\mathsf{pk}, w, \textsc{Sig}(\mathsf{sk}, w)) = 1] = 1. \tag{2.5}$$

The integer n is the *security parameter*.

This mathematical framework ensures authenticity and non-repudiation of the sender, and integrity of the message for a secure choice of the 3-tuple. The algorithm $\textsc{Gen}$ is the *key generation step*, which outputs a key pair as a pair of bit strings. The algorithm $\textsc{Sig}$ is the *signature generation step* and outputs a signature uniquely determined by the combination of the private key and the message. The algorithm $\textsc{Ver}$ is the *signature verification step* and checks if the message was signed correctly with the corresponding key, outputting a truth value.

We note that public keys in multivariate cryptography are polynomial systems with no apparent structure. However, if a malicious party can efficiently solve such systems, it can perform signature forgery, defeating the security of the underlying scheme. It is then crucial that solving random instances of polynomial systems is computationally hard, such that a cryptosystem can be based upon this problem. Fortunately, this is the case for several classes of polynomial systems.

Definition 2.2.4. Given a polynomial system of the form presented in Definition 2.1.20, finding a solution vector $x_1, \ldots, x_n$ is the *polynomial system solving problem*, or PoSSo.

Theorem 2.2.5 ((PATARIN; GOUBIN, 1997, App. A))**.** *Consider a polynomial system of* quadratic equations over $\mathbb{F}_2$ *of the form presented in Definition 2.1.20, then finding a solution vector $x_1, \ldots, x_n$ is NP-complete. This is the* multivariate quadratic (MQ) problem.

Hence, multivariate quadratic equations are good candidates for building cryptosystems based on Definition 2.2.4. Indeed, while the choice of $\mathbb{F}_q$ is more varied, most modern multivariate cryptography schemes are based on quadratic equations for efficiency reasons. We expand Definition 2.1.22 to its explicit form, taking into account the proposed degree restriction of the polynomials, to better visualize such systems.

Definition 2.2.6. Given $n,m \in \mathbb{N}$, a polynomial ring $\mathsf{F} = \mathbb{F}[x_1,\ldots,x_n]$ and $f^{(1)},\ldots,f^{(m)} \in \mathsf{F}$, a *system of multivariate quadratic polynomials over* $\mathbb{F}$ is defined as

$$\begin{cases} f^{(1)}(x_1,\ldots,x_n) & = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{ij}^{(1)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(1)} x_i + \delta^{(1)}, \\ \vdots & \\ f^{(m)}(x_1,\ldots,x_n) & = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{ij}^{(m)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(m)} x_i + \delta^{(m)}, \end{cases} \tag{2.6}$$

where $\alpha_{ij}^{(k)}, \beta_i^{(k)}, \delta^{(k)} \in \mathbb{F}$, $1 \le k \le m$. The set of all such systems is denoted by $\mathsf{MQ}(n,m)$. If all linear and constant terms are zero, then the resulting system is *homogeneous quadratic*, and the respective set is $\mathsf{MHQ}(n,m)$.

We recall that division of polynomials in a multivariate polynomial ring is ambiguous if a structure to order monomials is not given. More precisely, the quotient and remainder of a division between two multivariate polynomials depend on which is considered the divisor, that is, one representing the "largest monomial". Thus, we give a short discussion on how to order multivariate polynomials. The definition below amounts to ordering monomials according to their associated n-tuples of exponents, which is relatively intuitive.

Definition 2.2.7. Let $\mathbb{Z}^n$ be the set of n-tuples of integers and any $\alpha,\beta,\delta \in \mathbb{Z}^n$. Given a well-order $\prec$ in $\mathbb{Z}^n$ such that $\alpha+\delta \prec \beta+\delta$ if $\alpha \prec \beta$, a *monomial ordering* is defined by using α as $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \in \mathsf{R}[\mathbf{x}]$. Examples of monomial orderings are given below.

- The ordering given by $\mathbf{x}^\alpha \prec_{\mathsf{lex}} \mathbf{x}^\beta$ if $\alpha_i \prec \beta_i$ with minimal $1 \le i \le n$ such that $\alpha_i \neq \beta_i$ is the *lexicographic ordering*, or "dictionary ordering".

- The ordering given by $\mathbf{x}^\alpha \prec_{\mathsf{grlex}} \mathbf{x}^\beta$ if $\deg \mathbf{x}^\alpha \prec \deg \mathbf{x}^\beta$ or, as a tie-breaker, $\mathbf{x}^\alpha \prec_{\mathsf{lex}} \mathbf{x}^\beta$ is the *graded lexicographic ordering*.

- The ordering given by $\mathbf{x}^\alpha \prec_{\mathsf{grevlex}} \mathbf{x}^\beta$ if $\deg \mathbf{x}^\alpha \prec \deg \mathbf{x}^\beta$ or, as a tie-breaker, the last non-zero element of $\alpha - \beta$ is negative is the *graded reverse lexicographic ordering*.

In the context of multivariate cryptography, monomial ordering is used, for instance, to define matrix representations of multivariate polynomial systems, and to solve those using Gröbner bases (GATHEN; GERHARD, 2013, Sec. 21.4). Additionally, as seen in Section 1.3, the visualization and manipulation of multivariate polynomials as matrices are pivotal in understanding and even attacking some cryptosystems. Thus, we also show how to represent multivariate quadratic polynomials with or without constant or linear terms.

Definition 2.2.8. Given a polynomial system $\mathcal{P} \in \mathsf{MQ}(n,m)$, the *matrix representation of a polynomial* $f^{(i)} \in \mathcal{P}, 1 \leq i \leq m$ is given by

$$\overline{M}(f^{(i)}) = \begin{bmatrix} \alpha_{11}^{(i)} & \alpha_{12}^{(i)} & \cdots & \alpha_{1n}^{(i)} & \beta_1^{(i)} \\ 0 & \alpha_{22}^{(i)} & \cdots & \alpha_{2n}^{(i)} & \beta_2^{(i)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha_{nn}^{(i)} & \beta_n^{(i)} \\ 0 & 0 & \cdots & 0 & \delta^{(i)} \end{bmatrix} \tag{2.7}$$

such that $f^{(i)} = \mathbf{x} \cdot \overline{M}(f^{(i)}) \cdot \mathbf{x}^{\mathsf{T}}$ for $\mathbf{x} = (x_1, \ldots, x_n, 1)$.

Definition 2.2.9. Given a polynomial system $\mathcal{P} \in \mathsf{MHQ}(n,m)$, the *matrix representation of a homogeneous quadratic polynomial* $f^{(i)} \in \mathcal{P}, 1 \leq i \leq m$ is given by

$$M(f^{(i)}) = \begin{bmatrix} \alpha_{11}^{(i)} & \cdots & \alpha_{1n}^{(i)} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_{nn}^{(i)} \end{bmatrix} \tag{2.8}$$

such that $f^{(i)} = \mathbf{x} \cdot M(f^{(i)}) \cdot \mathbf{x}^{\mathsf{T}}$ for $\mathbf{x} = (x_1, \ldots, x_n)$. Its unique *symmetric* version is given by $\widetilde{M}(f^{(i)}) = M(f^{(i)}) + M(f^{(i)})^{\mathsf{T}}$.

While the matrix representation of $\mathcal{P}$ itself is highly dependant on a monomial ordering, the matrix of Definition 2.2.8 does not necessarily need any. However, the resulting product may differ from the original polynomial in this aspect. Further, we remark that the symmetric matrix of Definition 2.2.9 has a diagonal of zeroes if the underlying field characteristic is even.

We now turn to the most used framework to build cryptosystems in multivariate cryptography. The *bipolar construction*, due to its twofold nature, yields the possibility of building public-key encryption or signature schemes with the same idea.

Consider $n, m \in \mathbb{N}$ the number of variables and equations as in Definition 2.2.6, and $\mathcal{C} : \mathbb{F}^n \to \mathbb{F}^m \in \mathsf{MQ}(n,m)$. This is the *central map* and regarded as the trapdoor, that is, a function that is easy to compute but difficult to invert without particular information. Evidently, it shall be hidden to prevent signature forgery. Two *lateral maps* $\mathcal{L}_1 : \mathbb{F}^m \to \mathbb{F}^m \in \mathsf{GL}(\mathbb{F}^m)$ and $\mathcal{L}_2 : \mathbb{F}^n \to \mathbb{F}^n \in \mathsf{GL}(\mathbb{F}^n)$ are used to hide the structure of $\mathcal{C}$, by creating the composition $\mathcal{P} = \mathcal{L}_1 \circ \mathcal{C} \circ \mathcal{L}_2$ that looks random, making recovery of the central map hard.

The sequence of operations in the bipolar construction is summarized in Figure 1. It depicts the steps used to operate on a message $\mathbf{h}$ or ciphertext $\mathbf{z}$. As a consequence of the degree choice for multivariate cryptography, this is the only possible arrangement of maps. In other words, there always must be a quadratic trapdoor hidden by linear maps such that the composition of the functions is still a quadratic map. Further, the symbol $\mathcal{C}'$ is an abuse of language that means precisely the act of obtaining the preimage $\mathcal{C}(\mathbf{x}) = \mathbf{y}$. Since usually $m \neq n$, by definition, the central map cannot be invertible, and thus using the symbol $\mathcal{C}^{-1}$ could be misleading.

decryption / signature generation

$$\begin{tikzcd}
\mathbf{h} \in \mathbb{F}^m \arrow[r, "\mathcal{L}_1^{-1}"] & \mathbf{y} \in \mathbb{F}^m \arrow[r, "\mathcal{C}'"] & \mathbf{x} \in \mathbb{F}^n \arrow[r, "\mathcal{L}_2^{-1}"] & \mathbf{z} \in \mathbb{F}^n \arrow[lll, bend left, "\mathcal{P}"]
\end{tikzcd}$$

private

public

encryption / signature verification

Figure 1 – Standard flow of operations in the bipolar construction.

Source – the author.

We now observe the relationship between m and n to understand how the bipolar construction works. The lateral maps need not be considered in this discussion due to their lack of structure. For $m \geq n$, the map $\mathcal{C}$ should be injective, to ensure that there exists a unique plaintext $\mathbf{z}$ related to each ciphertext $\mathbf{h}$, by the preimage step. In the case of $m \leq n$, one expects that $\mathcal{C}$ is surjective, such that every message $\mathbf{h}$ has a signature $\mathbf{z}$, once again by the central map preimage.

The private and public keys are respectively a 3-tuple and its aforementioned composition, that is,

$$\begin{aligned} \mathsf{sk} &= (\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2) \in \mathsf{GL}(\mathbb{F}^m) \times \mathsf{MQ}(n,m) \times \mathsf{GL}(\mathbb{F}^n), \\ \mathsf{pk} &= \mathcal{P} \in \mathsf{MQ}(n,m). \end{aligned} \tag{2.9}$$

The bipolar construction can be extended to work differently from the main idea given, for instance in the case of schemes that perform calculations over multiple finite fields. We invite the reader to consult (PETZOLDT, 2013, Sec. 2.2) for a discussion of such schemes, and even ones that do not use the construction at all.

While very elegant, schemes derived from the bipolar construction are not exclusively based on Definition 2.2.4. There exist signature schemes constructed via the Fiat–Shamir transformation that only depend on PoSSo, such as MQDSS (CHEN et al., 2016), but their efficiency is lacking compared to schemes based on the bipolar construction. Consequently, a scheme using the bipolar construction is based on some variant of the *isomorphism of polynomials (IP) problem*. There are several versions of the IP problem, and we give a short definition for the variant that Oil–Vinegar schemes are based on, as introduced in (DING et al., 2008).

Definition 2.2.10. Let $\overline{\mathsf{MQ}}(n,m)$ be a class of structured quadratic polynomial maps and $\mathbb{U} = \mathsf{GL}(\mathbb{F}^m) \times \overline{\mathsf{MQ}}(n,m) \times \mathsf{GL}(\mathbb{F}^n)$ be a set of private keys. Given a 3-tuple $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2) \in \mathbb{U}$ with $\mathcal{P} = \mathcal{L}_1 \circ \mathcal{C} \circ \mathcal{L}_2$, finding a decomposition $\mathcal{P} = \widetilde{\mathcal{L}}_1 \circ \widetilde{\mathcal{C}} \circ \widetilde{\mathcal{L}}_2$ such that $(\widetilde{\mathcal{L}}_1, \widetilde{\mathcal{C}}, \widetilde{\mathcal{L}}_2) \in \mathbb{U}$ is the *extended isomorphism of polynomials (EIP) problem*.

The amount of information regarding the difficulty of EIP is sparse (THOMAE, 2013, p. 69), preventing complete security proofs for Oil–Vinegar schemes. This is also the case for other IP problems, and thus most multivariate cryptosystems simply have *ad hoc* security. A consequence of EIP is that there exist many private keys that compose a single public key (WOLF; PRENEEL, 2011), and this has been used to attack several schemes, as seen in Section 1.3.

Definition 2.2.11. Given $\mathbb{U}$ as per Definition 2.2.10, and private keys $(\mathcal{L}_1,\mathcal{C},\mathcal{L}_2),(\widetilde{\mathcal{L}}_1,\widetilde{\mathcal{C}},\widetilde{\mathcal{L}}_2) \in \mathbb{U}$, they are *equivalent keys* if

$$\mathcal{P} = \mathcal{L}_1 \circ \mathcal{C} \circ \mathcal{L}_2 = \widetilde{\mathcal{L}}_1 \circ \widetilde{\mathcal{C}} \circ \widetilde{\mathcal{L}}_2. \tag{2.10}$$

Definition 2.2.12. Given a private key $(\mathcal{L}_1,\mathcal{C},\mathcal{L}_2) \in \mathbb{U}$, linear transformations $\tau \in \mathsf{GL}(\mathbb{F}^m)$, $\omega \in \mathsf{GL}(\mathbb{F}^n)$ are *sustaining transformations* if they preserve the structure of the central map, that is,

$$\mathcal{P} = (\mathcal{L}_1 \circ \tau^{-1}) \circ (\tau \circ \mathcal{C} \circ \omega) \circ (\omega^{-1} \circ \mathcal{L}_2) \tag{2.11}$$

such that $\tau \circ \mathcal{C} \circ \omega \in \overline{\mathsf{MQ}}(n,m)$.

This concept is further explored in the context of the schemes of Section 2.3. Finally, we remark that our description of the bipolar construction is slightly different from the literature. The usual, broader description has $\mathcal{L}_1 \in \mathsf{AGL}(\mathbb{F}^m)$ and $\mathcal{L}_2 \in \mathsf{AGL}(\mathbb{F}^n)$. We recall that this is the *affine group*, which is composed of matrix-vector pairs $(M,\mathbf{b})$ that encode the affine transformation $\mathbf{a} \mapsto M\mathbf{a}+\mathbf{b}$. By setting the translation vector $\mathbf{b} = (0,\dots,0)$, we get a linear transformation.

However, it is argued in (WOLF; PRENEEL, 2011, Remark 3.1) that due to the theory of equivalent keys, it is not relevant to the security of most schemes based on the bipolar construction if $\mathcal{L}_1$ and $\mathcal{L}_2$ are linear instead of affine transformations. Hence, our description of the bipolar construction is based on this technical result. Furthermore, there is independent evidence of this fact for the schemes presented in Section 2.3.

It is argued in (THOMAE, 2013, Lemmas 3.2, 3.10, 3.19) that some IP problems modelled to represent Oil–Vinegar schemes are no harder to solve using linear transformations in place of affine ones. As a consequence, the polynomials in the central map must be homogeneous quadratic. Moreover, there exist works that confirm homogeneous quadratic maps and linear transformations are safe both in UOV (KIPNIS; SHAMIR, 1998, Sec. 5) (BRAEKEN; WOLF; PRENEEL, 2005, Sec. 3.1) and Rainbow (DING et al., 2019, Sec. 3.1).

2.3 THE OIL–VINEGAR TRAPDOOR

In this section, we describe the classic schemes based on the Oil–Vinegar trapdoor, introduced in (PATARIN, 1997). We define the Unbalanced Oil and Vinegar (UOV) signature scheme in Subsection 2.3.1, alongside with its multi-layer generalization Rainbow, in Subsection 2.3.2. We discuss the theory of equivalent keys for each scheme alongside their descriptions. Our definitions are based on (DING; GOWER; SCHMIDT, 2006, Chap. 3) and (PETZOLDT, 2013, Chap. 3).

2.3.1 Unbalanced Oil and Vinegar signature scheme

The idea of the original Oil and Vinegar signature scheme is to build the central map with two sets of variables, *vinegar and oil variables*, such that the multivariate quadratic polynomials do not feature any quadratic oil terms. Thus, by choosing the values of vinegar variables, the systems become linear in the oil variables and consequently easy to solve.

Definition 2.3.1. Given $v, o \in \mathbb{N}$, let $V = \{1, \ldots, v\}$, $O = \{v+1, \ldots, v+o\}$ and a vector $\mathbf{x} = (x_1, \ldots, x_v, x_{v+1}, \ldots, x_{v+o})$. Given a polynomial ring $\mathbb{F}[\mathbf{x}]$, an *Oil–Vinegar polynomial* is a multivariate quadratic polynomial of the form

$$\sum_{i \in V} \sum_{j \in V} \alpha_{ij} x_i x_j + \sum_{i \in V} \sum_{j \in O} \beta_{ij} x_i x_j + \sum_{i \in V \cup O} \gamma_i x_i + \delta, \tag{2.12}$$

where $\alpha_{ij}, \beta_{ij}, \gamma_i, \delta \in \mathbb{F}$. The set of all such polynomials is defined as $\mathsf{OVP}(v, o)$.

Definition 2.3.2. Let $n = v + o$. A polynomial map $\mathcal{C} : \mathbb{F}^n \to \mathbb{F}^o$ consisting of $f^{(1)}, \ldots, f^{(o)} \in \mathsf{OVP}(v, o)$ is an *Oil–Vinegar polynomial map*. The set of all such maps is defined as $\mathsf{OV}(n, o)$. If all polynomials are homogeneous quadratic, then the respective set is $\mathsf{OVH}(v, o)$.

The integers v and o represent the quantity of vinegar and oil variables, respectively. After the substitution of vinegar variables, the resulting system is exactly determined, with o equations and o variables. We remark that the first summation of Definition 2.3.1 is explicitly split into cross-vinegar terms, and product of vinegar and oil terms, to emphasize that there do not exist cross-oil terms. A visualization of this structure is given below.

Definition 2.3.3. Given a homogeneous quadratic Oil–Vinegar polynomial $f^{(i)} \in \mathsf{OVH}(v, o)$, its unique symmetric matrix representation is

$$\widetilde{M}(f^{(i)}) = \begin{bmatrix} \star_{v \times v} & \star_{v \times o} \\ \star_{o \times v} & \mathbf{0}_{o \times o} \end{bmatrix}. \tag{2.13}$$

We are now ready to describe the Oil–Vinegar signature scheme.

2.3.1.1 *Key generation.*

Since Oil–Vinegar schemes are based on the bipolar construction, the private key is composed of a central map $\mathcal{C}$ and a linear transformation $\mathcal{L}_2$ to hide its structure. We remark that the linear transformation $\mathcal{L}_1$ is always id, since it simply performs a permutation of the coefficients in the central map (THOMAE, 2013, p. 71) and is not relevant to the security of the scheme. Hence, we do not consider $\mathcal{L}_1$ to be part of the private key. Then, let

$$\mathsf{sk} = (\mathcal{C}, \mathcal{L}_2) \xleftarrow{\mathsf{r}} \mathsf{OV}(n, o) \times \mathsf{GL}(\mathbb{F}^n). \tag{2.14}$$

In the case of $\mathcal{C}$, the random choice amounts to picking all coefficients of the polynomials at random. For $\mathcal{L}_2$, a new matrix with random coefficients is generated until it is invertible.

As discussed in Section 2.2, we choose the lateral maps to be linear transformations, since it is proven that this also does not affect the security of Oil–Vinegar schemes (BRAEKEN; WOLF; PRENEEL, 2005, Sec. 3.1). The same fact holds for homogeneous quadratic Oil–Vinegar polynomials. Let $\mathcal{P} = \mathcal{C} \circ \mathcal{L}_2$ and

$$\mathsf{pk} = \mathcal{P} \in \mathsf{MQ}(n,o). \tag{2.15}$$

2.3.1.2 Signature generation.

Consider a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}^o$, any message $M \in \{0,1\}^*$, and obtain the message digest $\mathbf{h} = \mathcal{H}(M)$. The Oil–Vinegar trapdoor is built to allow the signer to randomly choose vinegar variables $x_1,\ldots,x_v$ such that a preimage of the message digest can be obtained, by solving for the remaining oil variables $\mathcal{C}(x_1,\ldots,x_v,x_{v+1},\ldots,x_{v+o}) = \mathbf{h}$.

To sign a message, one chooses $(x_1,\ldots,x_v) \overset{\mathsf{r}}{\leftarrow} \mathbb{F}^v$, substitutes these values into $\mathcal{C}(\mathbf{x}) = \mathbf{h}$ and solves the resulting linear system with, for instance, Gaussian elimination, obtaining the remaining values of $\mathbf{x}$. Usual parameters for Oil–Vinegar schemes are small enough, as per Subsection 3.3, such that special methods for solving linear systems are not needed.

If the linear system is inconsistent due to a bad choice of vinegar variables, new ones are simply picked. This event happens with a very low probability, as seen in Subsection 3.2.1. Then, the signature is the result of the inverse linear transformation applied to the preimage. In other words, $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$.

2.3.1.3 Signature verification.

To verify a signature, compute $\mathbf{h}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{h} = \mathbf{h}'$, then the signature is valid, and otherwise invalid.

The balanced Oil–Vinegar scheme occurs when $v = o$. However, it was found to be insecure shortly after its proposal (KIPNIS; SHAMIR, 1998). We discuss this attack in Chapter 4. In the following year, the UOV variant with $v = 2o$ was proposed as a secure variant (KIPNIS; PATARIN; GOUBIN, 1999), and no security weaknesses are known to the date of writing if the correct choice of parameters is performed. An instance of UOV is denoted by $\mathsf{UOV}(\mathbb{F}, v, o)$.

The size of a private key is the number of elements in $\psi(\mathcal{L}_2)$, added to the count of all valid monomial combinations in $\mathcal{C}$. In other words,

$$\#\mathsf{sk} = \underbrace{o \cdot \left(\frac{v \cdot (v+1)}{2} + v \cdot o + v + o + 1 \right)}_{\mathcal{C}} + \underbrace{n^2}_{\mathcal{L}_2} \tag{2.16}$$

field elements. The size of a public key is the number of coefficients of o polynomials in the ring $\mathbb{F}[x_1,\ldots,x_n]$, which is known to be $\binom{n+d}{d}$ for the degree $d \in \mathbb{N}$. Since Oil–Vinegar polynomials are quadratic, we have $d = 2$, and hence

$$\#\mathsf{pk} = o \cdot \frac{(n+1) \cdot (n+2)}{2} \tag{2.17}$$

field elements. We note that if the Oil–Vinegar polynomials are homogeneous quadratic, then the private key size decreases by $o \cdot (v+o+1)$ elements, and the public key size decreases by $o \cdot (n+1)$ elements.

2.3.1.4 *Equivalent keys.*

We recall that there exists a specific form of sustainers of Definition 2.2.12 for each scheme. In the case of UOV, these mappings are given in (WOLF; PRENEEL, 2011, Sec. 4.3). We follow the notation of (PETZOLDT, 2013).

Theorem 2.3.4 ((WOLF; PRENEEL, 2011, Thm. 4.15))**.** *Given a UOV private key* $(\mathcal{C}, \mathcal{L}_2) \in \mathsf{OV}(n,o) \times \mathsf{GL}(\mathbb{F}^n)$ *and* $\omega \in \mathsf{GL}(\mathbb{F}^n)$ *such that*

$$\psi(\omega) = \begin{bmatrix} \omega^{(1)}_{v\times v} & \mathbf{0}_{v\times o} \\ \omega^{(2)}_{o\times v} & \omega^{(3)}_{o\times o} \end{bmatrix} \tag{2.18}$$

with $\omega^{(1)} \in \mathsf{GL}_v(\mathbb{F})$ *and* $\omega^{(3)} \in \mathsf{GL}_o(\mathbb{F})$*, then* $(\mathcal{C} \circ \omega, \omega^{-1} \circ \mathcal{L}_2)$ *is an equivalent private key.*

We remark that $\psi(\omega^{-1})$ has the same form as ω due to the restrictions on $\omega^{(1)}$ and $\omega^{(3)}$, that is,

$$\psi(\omega^{-1}) = \begin{bmatrix} \left(\omega^{(1)}\right)^{-1} & \mathbf{0}_{v\times o} \\ -\left(\omega^{(3)}\right)^{-1} \cdot \omega^{(2)} \cdot \left(\omega^{(1)}\right)^{-1} & \left(\omega^{(3)}\right)^{-1} \end{bmatrix}, \tag{2.19}$$

where we remove the sub-indices that indicate the dimension of the block matrices to improve readability. It is a direct consequence of this theorem that there exist at least as many equivalent keys as possible choices for ω. This redundancy may be cast away by observing the shape of $\omega^{-1} \circ \mathcal{L}_2$ and only selecting lateral maps with this structure.

Theorem 2.3.5 ((BEULLENS, 2017, Thm. 2))**.** *Given a UOV private key* $(\mathcal{C}, \mathcal{L}_2) \in \mathsf{OV}(n,o) \times \mathsf{GL}(\mathbb{F}^n)$ *and* $\omega \in \mathsf{GL}(\mathbb{F}^n)$ *such that*

$$\psi(\mathcal{L}_2) = \begin{bmatrix} \mathcal{L}_2^{(1)} & \mathcal{L}_2^{(2)} \\ \mathcal{L}_2^{(3)} & \mathcal{L}_2^{(4)} \end{bmatrix}, \ \psi(\omega) = \begin{bmatrix} \mathcal{L}_2^{(1)} & \mathbf{0}_{v\times o} \\ \mathcal{L}_2^{(3)} & \mathcal{L}_2^{(4)} - \mathcal{L}_2^{(3)} \cdot \left(\mathcal{L}_2^{(1)}\right)^{-1} \cdot \mathcal{L}_2^{(2)} \end{bmatrix} \tag{2.20}$$

and $\mathcal{L}_2^{(1)} \in \mathsf{GL}_v(\mathbb{F})$*, then*

$$\psi(\omega^{-1} \circ \mathcal{L}_2) = \begin{bmatrix} \mathbf{I}_v & \left(\mathcal{L}_2^{(1)}\right)^{-1} \cdot \mathcal{L}_2^{(2)} \\ \mathbf{0}_{o\times v} & \mathbf{I}_o \end{bmatrix}. \tag{2.21}$$

Then, as long as $\mathcal{L}_2^{(1)}$ is invertible, there exists an equivalent key for UOV. This happens with overwhelming probability as per Subsection 3.2.1. Hence, one may simply choose $\widetilde{\mathcal{L}}_2 \in \mathsf{GL}(\mathbb{F}^n)$ such that

$$\psi(\widetilde{\mathcal{L}}_2) = \begin{bmatrix} \mathbf{I}_v & \star_{v\times o} \\ \mathbf{0}_{o\times v} & \mathbf{I}_o \end{bmatrix} \tag{2.22}$$

and obtain a reduction in the private key size of a factor $\frac{n^2}{vo}$.

2.3.2 Rainbow signature scheme

The Rainbow signature scheme, originally defined in (DING; SCHMIDT, 2005) and a finalist of the NIST standardization process (ALAGIC et al., 2020, Sec. 3.20), is a generalized version of UOV that reduces the length of keys and signatures. Several smaller "oil and vinegar" layers are combined to create a "rainbow", enabling this optimization.

Consider a finite field $\mathbb{F}$ and $u,n \in \mathbb{N}$ where $u \leq n$. Choose a sequence of integers $v_1,\dots,v_u$ such that $0 = v_0 < v_1 < \cdots < v_u < v_{u+1} = n$. Take the usual set $V = \{1,\dots,n\}$ and, for the range $1 \leq l \leq u$, define vinegar variables as $V_\ell = \{1,\dots,v_\ell\}$ and oil variables as $O_\ell = \{v_\ell + 1,\dots,v_{\ell+1}\}$. We note that $V_1 \subset \cdots \subset V_{u+1} = V$ and $O_\ell = V_{\ell+1} \setminus V_\ell$. Let $o_\ell = |O_\ell|$ and $m = n - v_1$. Now, we define u vector spaces spanned by quadratic Oil–Vinegar polynomials of the form $\mathsf{OV}(v_\ell,o_\ell)$, that is,

$$\sum_{i\in V_\ell}\sum_{j\in V_\ell}\alpha_{ij}x_ix_j + \sum_{i\in V_\ell}\sum_{j\in O_\ell}\beta_{ij}x_ix_j + \sum_{i\in V_\ell\cup O_\ell}\gamma_ix_i + \delta, \tag{2.23}$$

where $\alpha_{ij},\beta_{ij},\gamma_i,\delta \in \mathbb{F}$. We define $\mathsf{RB}(v_1,o_1,\dots,o_u) = \mathsf{OV}(v_1,o_1) \times \cdots \times \mathsf{OV}(v_u,o_u)$. We are now ready to describe the Rainbow signature scheme.

2.3.2.1 *Key generation.*

In the case of Rainbow, all three maps are needed to securely hide the structure of the central map. Therefore,

$$\mathsf{sk} = (\mathcal{L}_1,\mathcal{C},\mathcal{L}_2) \overset{\mathsf{r}}{\leftarrow} \mathsf{GL}(\mathbb{F}^m) \times \mathsf{RB}(v_1,o_1,\dots,o_u) \times \mathsf{GL}(\mathbb{F}^n). \tag{2.24}$$

For the central map $\mathcal{C}$, the random choice is equivalent to randomly choosing polynomial systems from each $\mathsf{OV}(v_\ell,o_\ell)$ with $1 \leq l \leq u$, analogous to multiple UOV systems. The linear transformations are random invertible matrices. Since each sequence of vinegar variables in a layer contains all oil and vinegar variables from the previous layer, this allows for the correct preimage calculation of $\mathcal{C}$. Let $\mathcal{P} = \mathcal{L}_1 \circ \mathcal{C} \circ \mathcal{L}_2$ and

$$\mathsf{pk} = \mathcal{P} \in \mathsf{MQ}(n,m). \tag{2.25}$$

For example, in the case of $u = 2$ and $1 \leq i \leq m$, the matrix representation of a homogeneous quadratic $\mathcal{C}$ is

$$\widetilde{M}(f^{(i)}) = \begin{cases} \begin{bmatrix} \star_{v_1\times v_1} & \star_{v_1\times o_1} & \mathbf{0}_{v_1\times o_2} \\ \star_{o_1\times v_1} & \mathbf{0}_{o_1\times o_1} & \mathbf{0}_{o_1\times o_2} \\ \mathbf{0}_{o_2\times v_1} & \mathbf{0}_{o_2\times o_1} & \mathbf{0}_{o_2\times o_2} \end{bmatrix} & \text{for } f^{(1)},\dots,f^{(o_1)} \in \mathcal{C}, \\ \\ \begin{bmatrix} \star_{v_1\times v_1} & \star_{v_1\times o_1} & \star_{v_1\times o_2} \\ \star_{o_1\times v_1} & \star_{o_1\times o_1} & \mathbf{0}_{o_1\times o_2} \\ \star_{o_2\times v_1} & \mathbf{0}_{o_2\times o_1} & \mathbf{0}_{o_2\times o_2} \end{bmatrix} & \text{for } f^{(o_1+1)},\dots,f^{(o_2)} \in \mathcal{C}. \end{cases} \tag{2.26}$$

2.3.2.2 *Signature generation.*

Consider a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}^m$, any message $M \in \{0,1\}^*$, and obtain the message digest $\mathbf{h} = \mathcal{H}(M)$. Compute $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$. To obtain the preimage $\mathcal{C}(\mathbf{x}) = \mathbf{y}$, every layer must be inverted recursively. One chooses $(x_1,\dots,x_{v_1}) \overset{r}{\leftarrow} \mathbb{F}^{v_1}$ and substitutes the values into the first layer, or in other words, the first system of o_1 polynomials. This brings forth a system of o_1 linear equations in $x_{v_1+1},\dots,x_{v_2}$. It can be solved with an algorithm such as Gaussian elimination. If the system does not have a solution, new vinegar variables $x_1,\dots,x_{v_1}$ have to be chosen. These solutions can then be substituted into the next layer, which creates a system of o_2 linear equations, that can be solved analogously. This procedure is repeated until all layers are solved. Finally, we compute $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$.

2.3.2.3 *Signature verification.*

To verify a signature, compute $\mathbf{h}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{h} = \mathbf{h}'$, then the signature is valid, and otherwise invalid.

The remarks about UOV regarding the size of parameters, chances of failure in the central map preimage calculation and lack of security weaknesses are maintained in the case of Rainbow. An instance of Rainbow is denoted by $\mathsf{Rainbow}(\mathbb{F}, v_1, o_1, \dots, o_u)$, with usually $u = 2$ for efficiency reasons. We note that when $u = 1$, we get the UOV scheme.

Measured in field elements, the size of a private key is the number of elements in $\psi(\mathcal{L}_1)$ and $\psi(\mathcal{L}_2)$, added to the count of all valid monomial combinations from each layer of Oil–Vinegar polynomials as per Equation (2.23). In other words,

$$\#\mathsf{sk} = \underbrace{m^2}_{\mathcal{L}_1} + \underbrace{\sum_{k=1}^{u} o_k \cdot \left(\frac{v_k \cdot (v_k+1)}{2} + v_k \cdot o_k + v_k + o_k + 1\right)}_{\mathcal{C}} + \underbrace{n^2}_{\mathcal{L}_2}. \tag{2.27}$$

The size of a public key is similar to the UOV case, and thus we have

$$\#\mathsf{pk} = m \cdot \frac{(n+1)\cdot(n+2)}{2} \tag{2.28}$$

field elements. If polynomials are homogeneous quadratic, then the private key size decreases by $\sum_{k=1}^{u} o_k \cdot (v_k + o_k + 1)$ elements, and the public key size decreases by $m \cdot (n+1)$ elements.

2.3.2.4 *Equivalent keys.*

As a generalization of UOV, there exist equivalent keys in Rainbow using a similar rationale. The theory is developed in (WOLF; PRENEEL, 2011, Sec. 4.4) for STS, a particular sparse case of Rainbow. It is argued in (PETZOLDT, 2013) that the strategy is also suitable for Rainbow.

Theorem 2.3.6 ((WOLF; PRENEEL, 2011, Thm. 4.17))**.** *Without loss of generality, consider a private key $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2) \in \mathsf{GL}(\mathbb{F}^m) \times \mathsf{RB}(v_1, o_1, \ldots, o_u) \times \mathsf{GL}(\mathbb{F}^n)$ with $u = 2$. Given $\tau \in \mathsf{GL}(\mathbb{F}^m)$ and $\omega \in \mathsf{GL}(\mathbb{F}^n)$ such that*

$$\psi(\tau) = \begin{bmatrix} \tau^{(1)}_{o_1 \times o_1} & \mathbf{0}_{o_1 \times o_2} \\ \tau^{(2)}_{o_2 \times o_1} & \tau^{(3)}_{o_2 \times o_2} \end{bmatrix}, \ \psi(\omega) = \begin{bmatrix} \omega^{(1)}_{v_1 \times v_1} & \mathbf{0}_{v_1 \times o_1} & \mathbf{0}_{v_1 \times o_2} \\ \omega^{(2)}_{o_1 \times v_1} & \omega^{(3)}_{o_1 \times o_1} & \mathbf{0}_{o_1 \times o_2} \\ \omega^{(4)}_{o_2 \times v_1} & \omega^{(5)}_{o_2 \times o_1} & \omega^{(6)}_{o_2 \times o_2} \end{bmatrix}, \tag{2.29}$$

$\omega^{(1)} \in \mathsf{GL}_{v_1}(\mathbb{F})$, $\tau^{(1)}, \omega^{(3)} \in \mathsf{GL}_{o_1}(\mathbb{F})$ and $\tau^{(3)}, \omega^{(6)} \in \mathsf{GL}_{o_2}(\mathbb{F})$, then $(\mathcal{L}_1 \circ \tau^{-1}, \tau \circ \mathcal{C} \circ \omega, \omega^{-1} \circ \mathcal{L}_2)$ is an equivalent private key.

The linear transformations τ^{-1} and ω^{-1} maintain their forms by a similar block inverse calculation, as seen below. We use a compact notation to represent operations between inner block matrices for easier reading. For example, $-(\tau^{(3)})^{-1} \cdot \tau^{(2)} \cdot (\tau^{(1)})^{-1}$ is denoted as $\tau^{(-3^{-1} \cdot 2 \cdot 1^{-1})}$. Then,

$$\begin{aligned} \psi(\tau^{-1}) &= \begin{bmatrix} (\tau^{(1)})^{-1} & \mathbf{0}_{o_1 \times o_2} \\ \tau^{(-3^{-1} \cdot 2 \cdot 1^{-1})} & (\tau^{(3)})^{-1} \end{bmatrix}, \\ \psi(\omega^{-1}) &= \begin{bmatrix} (\omega^{(1)})^{-1} & \mathbf{0}_{v_1 \times o_1} & \mathbf{0}_{v_1 \times o_2} \\ \omega^{(-3^{-1} \cdot 2 \cdot 1^{-1})} & (\omega^{(3)})^{-1} & \mathbf{0}_{o_1 \times o_2} \\ \omega^{6^{-1} \cdot (4 + 5 \cdot 3^{-1} \cdot 2) \cdot 1^{-1}} & \omega^{(-6^{-1} \cdot 5 \cdot 3^{-1})} & (\omega^{(6)})^{-1} \end{bmatrix}. \end{aligned} \tag{2.30}$$

The redundancy is removed by the same rationale of UOV, observing the shape of the lateral maps in the equivalent key.

Theorem 2.3.7 ((PETZOLDT, 2013, Thm. 3.6))**.** *Without loss of generality, consider a private key $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2) \in \mathsf{GL}(\mathbb{F}^m) \times \mathsf{RB}(v_1, o_1, \ldots, o_u) \times \mathsf{GL}(\mathbb{F}^n)$ with $u = 2$, such that*

$$\psi(\mathcal{L}_1) = \begin{bmatrix} \mathcal{L}_1^{(1)} & \mathcal{L}_1^{(2)} \\ \mathcal{L}_1^{(3)} & \mathcal{L}_1^{(4)} \end{bmatrix} \text{ and } \psi(\mathcal{L}_2) = \begin{bmatrix} \mathcal{L}_2^{(1)} & \mathcal{L}_2^{(2)} & \mathcal{L}_2^{(3)} \\ \mathcal{L}_2^{(4)} & \mathcal{L}_2^{(5)} & \mathcal{L}_2^{(6)} \\ \mathcal{L}_2^{(7)} & \mathcal{L}_2^{(8)} & \mathcal{L}_2^{(9)} \end{bmatrix}. \tag{2.31}$$

1. *Given $\tau \in \mathsf{GL}(\mathbb{F}^m)$ such that*

$$\psi(\tau) = \begin{bmatrix} \mathcal{L}_1^{(1 - 2 \cdot 4^{-1} \cdot 3)} & \mathbf{0}_{o_1 \times o_2} \\ \mathcal{L}_1^{(3)} & \mathcal{L}_1^{(4)} \end{bmatrix} \tag{2.32}$$

and $\mathcal{L}_1^{(4)} \in \mathsf{GL}_{o_2}(\mathbb{F})$, then

$$\psi(\mathcal{L}_1 \circ \tau^{-1}) = \begin{bmatrix} \mathbf{I}_{o_1} & \mathcal{L}_1^{(2 \cdot 4^{-1})} \\ \mathbf{0}_{o_2 \times o_1} & \mathbf{I}_{o_2} \end{bmatrix}. \tag{2.33}$$

2. *Given* $\omega \in \mathsf{GL}(\mathbb{F}^n)$ *such that*

$$\psi(\omega) = \begin{bmatrix} \mathcal{L}_2^{(1)} & \mathbf{0}_{v_1 \times o_1} & \mathbf{0}_{v_1 \times o_2} \\ \mathcal{L}_2^{(4)} & \mathcal{L}_2^{(5-4\cdot 1^{-1}\cdot 2)} & \mathbf{0}_{o_1 \times o_2} \\ \mathcal{L}_2^{(7)} & \mathcal{L}_2^{(8-7\cdot 1^{-1}\cdot 2)} & \omega^{(6)} \end{bmatrix}, \tag{2.34}$$

$$\omega^{(6)} = \mathcal{L}_2^{((9-7\cdot 1^{-1}\cdot 3)-(8-7\cdot 1^{-1}\cdot 2)\cdot(5-4\cdot 1^{-1}\cdot 2)^{-1}\cdot(6-4\cdot 1^{-1}\cdot 3))}$$

with $\mathcal{L}_2^{(1)} \in \mathsf{GL}_{v_1}(\mathbb{F})$ *and* $\omega^{(3)} \in \mathsf{GL}_{o_1}(\mathbb{F})$*, then*

$$\psi(\omega^{-1} \circ \mathcal{L}_2) = \begin{bmatrix} \mathbf{I}_{v_1} & \mathcal{L}_2^{(-1\cdot 2)} & \mathcal{L}_2^{(-1\cdot 3)} \\ \mathbf{0}_{o_1 \times v_1} & \mathbf{I}_{o_1} & \left(\omega^{(3)}\right)^{-1} \cdot \mathcal{L}_2^{(6-4\cdot 1^{-1}\cdot 3)} \\ \mathbf{0}_{o_2 \times v_1} & \mathbf{0}_{o_2 \times o_1} & \mathbf{I}_{o_2} \end{bmatrix}. \tag{2.35}$$

If $\mathcal{L}_2^{(1)}$ and $\omega^{(3)}$ are invertible, which once again happens with overwhelming probability, then there exists an equivalent key for Rainbow with $\widetilde{\mathcal{L}}_1 \in \mathsf{GL}(\mathbb{F}^m)$ and $\widetilde{\mathcal{L}}_2 \in \mathsf{GL}(\mathbb{F}^n)$ such that

$$\psi(\widetilde{\mathcal{L}}_1) = \begin{bmatrix} \mathbf{I}_{o_1} & \star_{o_1 \times o_2} \\ \mathbf{0}_{o_2 \times o_1} & \mathbf{I}_{o_2} \end{bmatrix}, \ \psi(\widetilde{\mathcal{L}}_2) = \begin{bmatrix} \mathbf{I}_{v_1} & \star_{v_1 \times o_1} & \star_{v_1 \times o_2} \\ \mathbf{0}_{o_1 \times v_1} & \mathbf{I}_{o_1} & \star_{o_1 \times o_2} \\ \mathbf{0}_{o_2 \times v_1} & \mathbf{0}_{o_2 \times o_1} & \mathbf{I}_{o_2} \end{bmatrix}. \tag{2.36}$$

The private key size is consequently reduced by a factor of $\frac{m^2}{o_1 o_2} + \frac{n^2}{v_1 o_1 + v_2 o_2}$.

There exists a generalization of the theory of equivalent keys, developed in (THOMAE, 2013), called *good keys*. It is the result of cryptanalytic developments that targeted a specific variant of Rainbow, which claimed to reduce private keys, with the intent of proving it insecure (THOMAE; WOLF, 2012). A particular consequence of the theory underlying this attack is that good keys may be used to further reduce private keys, through distinct linear transformations τ and ω.

More precisely, several quadratic terms in the central map $\mathcal{C}$ do not contribute to the security of the scheme, as seen in Section 3.1.2. The structure of the lateral maps in the context of good keys is depicted in (SHIM; PARK; KOO, 2020, Fig. 3). Specifically, for $u = 2$, $\psi(\widetilde{\mathcal{L}}_1)$ is $\mathbf{I}_m$, except for the last o_2 values of the o_1-th row, which are arbitrary. Analogously, $\psi(\widetilde{\mathcal{L}}_2)$ is $\mathbf{I}_n$, except for the arbitrary first v_2 values of its last column. We hereafter follow the more common usage of equivalent keys in the literature, but we encourage further research on the reduction of private keys using good keys.

3 NEW APPROACH TO REDUCE PRIVATE KEYS

We observe that the reuse of vinegar variables in the signature generation step of the Rainbow scheme leads to a shorter representation of its central map, and consequently, of the entire private key. We analyze the security implications of this strategy and present modifications to the Rainbow scheme, enabling a reduction in private key sizes of up to 85% with secure parameters. Furthermore, this framework can be applied on top of existing schemes that shorten private or public keys, spawning derivatives that reduce the total key pair size by a factor of 3.5.

The contents of this chapter are an extended version of (ZAMBONIN; BITTENCOURT; CUSTÓDIO, 2019). In Section 3.1, we give a formal description of our modifications, alongside with a security analysis taking into account current cryptanalytic methods for Rainbow. In Section 3.2, we argue that no change of parameters is needed to maintain performance and randomness when creating signatures. In Section 3.3, we measure how our modifications behave when applied to current sets of parameters for Rainbow-like schemes.

3.1 MODIFICATIONS TO THE ORIGINAL SCHEME

Our approach consists of modifications to the key and signature generation steps of Rainbow-like signature schemes. We propose to reuse the first set of vinegar variables for several signatures and replace these only when necessary. In other words, situations where the central map preimage cannot be calculated would fail. By locking such variables and substituting them on the central map $\mathcal{C}$ early in the key generation algorithm, we create a $\overline{\mathcal{C}}$ map linear in v_1, thus reducing storage requirements.

We recall that as per Subsection 1.3.1 most variants that shorten private keys are structural. Hence, some heuristic limits the key space with the intent of producing a compact private key. Moreover, the primary approach to reduce public keys (PETZOLDT, 2013) prevents alterations to the private key. It indirectly generates $\mathcal{C}$ from a partial public key through linear relations between the maps.

Alternatively, our proposal does not modify the underlying structure of the key pair, but rather of the central map preimages, and consequently of the signatures. We thus present three general methods based on different techniques that shorten private keys in all Rainbow-like schemes. We collectively denote these by Rainbow-η, as an easy way to refer to the general method, and use the same definitions as in Subsection 2.3.2, further denoting the vinegar variables for the first layer as $\widetilde{V}_1 = (x_1, \ldots, x_{v_1})$.

3.1.1 Usage of a seed

We use the fact that a PRNG can recreate the same sequence of numbers given a seed. The choice of such a generator is outside the scope of our work, and we assume that a cryptographically secure PRNG is chosen. This approach is similar to (SHIM; PARK; BAEK, 2015), but it is not as costly, since the private key does not need to be recreated before every signature generation. It is best suited to environments in which an efficient generator is previously supplied. We are not aware of any Rainbow variants that disallow this practice. Let us call this variant Rainbow-η_1.

3.1.1.1 Key generation.

We bound the creation of the key pair to a seed $\mathbf{S} \in \mathbb{N}_0$. Thus, $\mathcal{L}_1$, $\mathcal{C}$ and $\mathcal{L}_2$ are generated through the target scheme key generation algorithm, seeded by $\mathbf{S}$. The composition $\mathcal{P} = \mathcal{L}_1 \circ \mathcal{C} \circ \mathcal{L}_2$ must be obtained before the substitution of the vinegar variables such that it does not need to be recreated. We then choose $\widetilde{V}_1 \overset{r}{\leftarrow} \mathbb{F}^{v_1}$ and substitute these into $\mathcal{C}$, giving $\overline{\mathcal{C}} \in \mathsf{RB}(v_2, o_2, \ldots, o_u)$. Finally,

$$\begin{aligned} \mathsf{sk}^{\eta_1} &= (\mathbf{S}, \widetilde{V}_1, \mathcal{L}_1, \overline{\mathcal{C}}, \mathcal{L}_2), \\ \mathsf{pk}^{\eta_1} &= \mathcal{P}. \end{aligned} \tag{3.1}$$

3.1.1.2 Signature generation.

This step does not change significantly from the original scheme. Consider a message M and obtain its digest $\mathbf{h} = \mathcal{H}(M)$. Compute $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$. The preimage $\overline{\mathcal{C}}(\mathbf{x}) = \mathbf{y}$ already has its first v_1 elements set in the form of $\widetilde{V}_1$. The rest are obtained by recursively solving the remaining linear systems arising from the first layer. In the rare case that a failure occurs in the central map preimage calculation algorithm, we invoke the key generation step with $\mathbf{S}$ to obtain $\mathcal{C}$, choose other values for $\widetilde{V}_1$ and create a different $\overline{\mathcal{C}}$. Finally, compute $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$.

3.1.1.3 Signature verification.

This step does not change. To verify a signature, compute $\mathbf{h}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{h} = \mathbf{h}'$, then the signature is valid, and otherwise invalid.

3.1.2 Linear relations of Petzoldt

This approach is based on the fact that a private key owner can recover the original central map through the possession of the lateral maps and the public key. We use the linear relations first introduced in (PETZOLDT; BULYGIN; BUCHMANN, 2010a) and applied in the definition of the well-known CyclicRainbow scheme. We denote this variant as Rainbow-η_2. A brief explanation of the relations is given below, with the full rationale available in (PETZOLDT, 2013, Chap. 7).

Consider the key pair of a UOV signature scheme as described in Equations (2.14) and (2.15). It is known that quadratic terms do not mix with linear and constant terms in the case of Oil–Vinegar signature schemes, and thus we refer only to the quadratic part of any polynomial systems in our description. Let the quadratic part of the k-th polynomial $c^{(k)} \in \mathcal{C},\ 1 \leq k \leq m$, represented by Definition 2.3.1, be denoted as

$$\widetilde{c}^{(k)} = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{ij}^{(k)} x_i x_j. \tag{3.2}$$

Further, we analogously denote the quadratic part of $h^{(k)} \in \mathcal{P}$ as

$$\widetilde{h}^{(k)} = \sum_{i=1}^{n}\sum_{j=i}^{n} \nu_{ij}^{(k)} x_i x_j. \tag{3.3}$$

Let the elements of $\psi(\mathcal{L}_2)$ be denoted as $d_{ij}, 1 \leq i,j \leq n$. We recall that the function composition $\mathcal{P} = \mathcal{C} \circ \mathcal{L}_2$ can be represented as a series of matrix multiplication operations. For instance, Equation (3.3) may be rewritten as

$$\widetilde{h}^{(k)} = \sum_{i=1}^{n}\sum_{j=i}^{n} \left(\alpha_{ij}^{(k)} \sum_{r=1}^{n} d_{ri} x_r \sum_{s=1}^{n} d_{sj} x_s \right). \tag{3.4}$$

Focusing on the inner summation, we can represent the matrix multiplication by $\psi(\mathcal{L}_2)$ in a compact way by letting

$$d_{rs}^{(ij)} = \begin{cases} d_{ri} \cdot d_{rj} & \text{if } r = s, \\ d_{ri} \cdot d_{sj} + d_{si} \cdot d_{rj} & \text{otherwise,} \end{cases} \tag{3.5}$$

allowing the rewrite of Equation (3.4) as

$$\widetilde{h}^{(k)} = \sum_{i=1}^{n}\sum_{j=i}^{n} \left(\alpha_{ij}^{(k)} \sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^{(ij)} x_r x_s \right). \tag{3.6}$$

If we recall Equation (3.3), the coefficient associated with the monomial $x_i x_j$ of the public key is thus exactly

$$\nu_{ij}^{(k)} = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{ij}^{(k)} \sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^{(ij)}. \tag{3.7}$$

By fixing a specific $\mathcal{L}_2$, Equation (3.7) becomes a linear relation between the coefficients of the private and public keys. This allows for the reconstruction of the quadratic part of $\mathcal{C}$ from $\mathcal{P}$ and $\mathcal{L}_2$. Thus, if the scheme uses homogeneous quadratic polynomials, the entire central map can be reconstructed. In the literature, this is depicted by representing the coefficients of maps $\mathcal{C}$ and $\mathcal{P}$ as a matrix with respect to some monomial ordering, and analogously for $d_{rs}^{(ij)}$ (PETZOLDT, 2013, p. 94).

The process for Rainbow is similar, with a small difference due to the presence of two lateral maps in the private key. Equations (2.24) and (2.25) show, respectively, the private and public keys of Rainbow. Since $\mathcal{L}_1$ simply permutes coefficients maintaining the structure of $\mathcal{C}$, we let $\mathcal{Q} = \mathcal{C} \circ \mathcal{L}_2$, and consequently $\mathcal{P} = \mathcal{L}_1 \circ \mathcal{Q}$. The relationship between $\mathcal{Q}$ and $\mathcal{C}$ is exactly as described above in the case of UOV. Let $q_{ij}^{(k)}$ be the coefficient associated with the monomial $x_i x_j$ in the k-th component of $\mathcal{Q}$. Then,

$$q_{ij}^{(k)} = \sum_{i=1}^{n} \sum_{j=i}^{n} \alpha_{ij}^{(k)} \sum_{r=1}^{n} \sum_{s=1}^{n} d_{rs}^{(ij)}, \tag{3.8}$$

and by letting the elements of $\psi(\mathcal{L}_1)$ be denoted as $e_{kl}, 1 \leq k, l \leq m$, then

$$v_{ij}^{(k)} = \sum_{l=1}^{m} e_{kl} q_{ij}^{(k)}. \tag{3.9}$$

We note that the performance of this method is lower than that of Rainbow-η_1, due to the efficiency of PRNGs and the inherent amount of matrix multiplications to recreate $\mathcal{C}$. However, it is a general technique that works on all Rainbow-like schemes if the polynomials in the central map are homogeneous quadratic, or their linear and constant parts are independently stored.

3.1.2.1 Key generation.

The usual key generation algorithm for the target scheme is used, yielding $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2)$ and $\mathcal{P}$. Substitute the sequence $\widetilde{V}_1 \xleftarrow{r} \mathbb{F}^{v_1}$ into $\mathcal{C}$, giving $\overline{\mathcal{C}}$. Then,

$$\begin{aligned} \mathsf{sk}^{\eta_2} &= (\widetilde{V}_1, \mathcal{L}_1, \overline{\mathcal{C}}, \mathcal{L}_2), \\ \mathsf{pk}^{\eta_2} &= \mathcal{P}. \end{aligned} \tag{3.10}$$

3.1.2.2 Signature generation.

This step does not change significantly from the original scheme. Consider a message M and obtain its digest $\mathbf{h} = \mathcal{H}(M)$. Compute $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$, and attempt to generate the preimage $\overline{\mathcal{C}}(\mathbf{x}) = \mathbf{y}$ by inverting every layer recursively. By the relations above, one is able to reconstruct $\mathcal{C}$ with no additional mechanisms if any transitory systems are not solvable, choose other vinegar variables and create another $\overline{\mathcal{C}}$. Finally, compute $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$.

3.1.2.3 Signature verification.

This step does not change. If $\mathbf{h} = \mathcal{P}(\mathbf{z})$, then the signature is valid, and otherwise invalid.

3.1.3 Application to the EU-CMA variant

The second-round Rainbow submission to the NIST standardization process (DING et al., 2019) gives a scheme description that diverges from the original works. The authors introduce modifications that provide security against the EU-CMA model, whereas the original scheme only prevents a weaker notion of universal forgery. We first recall the target security model.

Definition 3.1.1 ((CHEN et al., 2016, Def. 1.2))**.** Let a digital signature scheme parametrized by $n \in \mathbb{N}$ be denoted by $\textsc{Dss}(1^n) = (\textsc{Gen}, \textsc{Sig}, \textsc{Ver})$. Given a probabilistic polynomial-time algorithm $\mathcal{A}$ which can perform $Q \in \mathbb{N}$ queries, usually called the *adversary*, let the following experiment be defined:

$$\begin{aligned}
&\textbf{Experiment } \mathsf{Exp}^{\text{EU-CMA}}_{\textsc{Dss}(1^n)}(\mathcal{A}) \\
&\quad (\mathsf{sk}, \mathsf{pk}) \leftarrow \textsc{Gen}(1^n) \\
&\quad (M', \mathbf{z}') \leftarrow \mathcal{A}^{\textsc{Sig}(\mathsf{sk}, \cdot)}(\mathsf{pk}) \\
&\quad \rho \leftarrow \{M_i : 1 \leq i \leq Q\} \text{ are the queries to } \textsc{Sig}(\mathsf{sk}, \cdot) \\
&\quad \text{Return 1 iff } \textsc{Ver}(\mathsf{pk}, M', \mathrm{z}') = 1 \text{ and } M' \notin \rho.
\end{aligned} \tag{3.11}$$

A signature scheme $\textsc{Dss}(1^n)$ is *existentially unforgeable under adaptive chosen message attacks (EU-CMA)* if $\mathcal{A}$ has negligible success probability in $\mathsf{Exp}^{\text{EU-CMA}}_{\textsc{Dss}(1^n)}(\mathcal{A})$.

Given a public key, the adversary can obtain signatures from at most Q messages. These messages may be chosen adaptively, that is, a new query may depend on replies to previous ones. The adversary succeeds if it produces a valid pair, consisting of a message not in a previous query, and a signature valid under pk. The EU-CMA model is considered the most strict notion of security for signatures. In the case of Rainbow, it is possible to provide security according to this model through small modifications to the scheme.

More precisely, the introduction of a "salt" prevents that two signatures of the same message are correlated. We briefly describe this approach, with the intent of preventing the recalculation of $\overline{\mathcal{C}}$ in the case that $\widetilde{V}_1$ is not suitable. Let us denote this method as Rainbow-η_3. Concrete implementations of this method are tested on Section 3.3.

3.1.3.1 *Key generation.*

Consider $w \in \mathbb{N}$ as the length of the aforementioned salt. Generate the maps $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2)$ and $\mathcal{P}$ through the usual key generation algorithm. Substitute the sequence $\widetilde{V}_1 \xleftarrow{\mathsf{r}} \mathbb{F}^{v_1}$ into $\mathcal{C}$, giving $\overline{\mathcal{C}}$. We remark that the approach of Rainbow-η_1 may be applied here, that is, using a seed $\mathbf{S}$ to yield the maps. Then,

$$\begin{aligned}
\mathsf{sk}^{\eta_3} &= (\widetilde{V}_1, \mathcal{L}_1, \overline{\mathcal{C}}, \mathcal{L}_2, w), \\
\mathsf{pk}^{\eta_3} &= (\mathcal{P}, w).
\end{aligned} \tag{3.12}$$

3.1.3.2 *Signature generation.*

Let $r \xleftarrow{\mathsf{r}} \{0,1\}^w$. Consider a message M and obtain its digest by calculating $\mathbf{h} = \mathcal{H}(\mathcal{H}(M) \mid\mid r)$. Compute $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$, and attempt to generate the preimage $\overline{\mathcal{C}}(\mathbf{x}) = \mathbf{y}$. In the rare case that this does not succeed, other variables in $\widetilde{V}_1$ would be chosen and a new $\overline{\mathcal{C}}$ created. However, the concatenation of a salt to the original message digest alters $\mathbf{h}$ completely, due to the application of $\mathcal{H}$. Thus, it is only necessary to generate a new r and restart the signature generation process, such that $\widetilde{V}_1$, and consequently $\overline{\mathcal{C}}$, need not be modified. Alternatively, if the preimage is generated successfully, we finish by computing $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$. The signature is the pair $\mathbf{z}' = (\mathbf{z}, r)$.

3.1.3.3 *Signature verification.*

We recall that r is used to obtain $\mathbf{h}$ again. Then, if $\mathbf{h} = \mathcal{P}(\mathbf{z})$, the signature is valid, and otherwise invalid.

We are now able to discuss private key sizes. We note that all coefficients related to monomials with variables in $\widetilde{V}_1$ are now removed from the central map. The public key size does not generally change, and it is evident that the methods mentioned above only slightly modify the total size. Thus, we first give a general proposition.

Proposition 3.1.2. *The size of a Rainbow-η private key, denoted by* sk^η*, is*

$$\#\mathsf{sk}^\eta = \underbrace{v_1}_{\widetilde{V}_1} + \underbrace{m^2}_{\mathcal{L}_1} + \underbrace{n^2}_{\mathcal{L}_2} + \underbrace{\sum_{k=1}^{u} o_k \cdot \left(\frac{(v_k - v_1)(v_k - v_1 + 1)}{2} + (v_k - v_1) \cdot o_k + (v_{k+1} - v_1) + 1 \right)}_{\overline{\mathcal{C}}}. \quad (3.13)$$

field elements. If the polynomials of $\overline{\mathcal{C}}$ *are homogeneous quadratic, the size decreases by* $\sum_{k=1}^{u} o_k \cdot (v_{k+1} - v_1 + 1)$ *elements.*

Proof. We subtract v_1 terms from every collection of v_k monomials in Equation (2.27), and add $\widetilde{V}_1$ so that the whole preimages can be computed. □

The private key size for Rainbow-η_1 is exactly $\#\mathsf{sk}^\eta$ plus the size of the seed $\mathbf{S}$, usually a standard integral data type composed of a small number of bytes, e.g. eight. In the case of Rainbow-η_2, the polynomials in the central map must be homogeneous quadratic. For Rainbow-η_3, the length of the salt is taken to be $w = 128$ bits (DING et al., 2019, p. 11), and thus only 7 bits, or the equivalent representation in a non-binary field, are added to $\#\mathsf{sk}^\eta$. We note that the public key size of Rainbow-η_3 also increases by this amount.

3.1.4 Security analysis

A variety of attacks currently thwart the security of Rainbow-like signature schemes if parameters are not chosen carefully. We briefly state each attack, its estimated complexity, and argue that our methods do not facilitate such a strategy. Moreover, we also mention attacks which do not target the structure of Rainbow, but a concrete implementation of the scheme or one of its building blocks.

We follow the analysis of (DING et al., 2019, Sec. 8) and (PETZOLDT; BULYGIN; BUCHMANN, 2010c) for attacks in the classical setting. Direct and collision attacks aim for signature forgery, since there exist many possible $\mathbf{x}$ for any digest $\mathbf{h}$. Side-channel attacks are a general definition that may encompass either signature forgery or key recovery attacks. All other attacks are considered key recovery attacks.

3.1.4.1 Direct attack.

An attacker with possession of a digest $\mathbf{h}$ and the public key $\mathcal{P}$ tries to solve $\mathcal{P}(\mathbf{x}) = \mathbf{h}$. This is done by fixing some of the variables until the system is exactly determined and applying an algorithm built upon the theory of Gröbner bases, such as the Hybrid approach (BETTALE; FAUGÈRE; PERRET, 2012). While it is hard to pinpoint the exact running time of such methods, the authors estimate its asymptotic complexity as $2^{n\cdot\boldsymbol{\omega}(1.38-0.44\boldsymbol{\omega}\log_2(q)^{-1})}$ (BETTALE; FAUGÈRE; PERRET, 2012, Thm. 3), when $n = m$, for the linear algebra constant $2 \leq \boldsymbol{\omega} \leq 3$.

3.1.4.2 Collision attack.

Since only the message digest is signed in the case of multivariate signature schemes, the cryptographic hash function $\mathcal{H}$ used to calculate such digest must be secure against collision attacks. It is not within the scope of this work to consider attacks against specific functions, but we remark that these must also be taken into account. The message digest is an m-tuple of elements in $\mathbb{F}$, which has order q. We assume that elements of $\mathbb{F}$ are trivially represented as a collection of bits of length $\log_2 q$. Thus, one must require at least $\sqrt{2^{m\cdot\log_2 q}}$ evaluations of $\mathcal{H}$ to compute a collision, due to the birthday attack.

3.1.4.3 UOV attack.

The multi-layer approach of Rainbow does not hinder attacks that also work on the UOV signature scheme. Kipnis and Shamir created this attack (KIPNIS; SHAMIR, 1998) to break the balanced Oil–Vinegar scheme. The objective of this attack is to obtain an equivalent private key through finding the preimage of a specific oil subspace under the map $\mathcal{L}_2$. The complexity of the generalized attack for unbalanced schemes (KIPNIS; PATARIN; GOUBIN, 1999) is $q^{n-1-2\cdot o_u} \cdot o_u^4$ field multiplications. We discuss the UOV attack in more detail on Subsection 4.1.1.

3.1.4.4 MinRank attack (MR).

We recall that systems of polynomials in the public key $\mathcal{P}$ may be individually represented as matrices. This attack consists of finding o_1 linear combinations of these, such that they have rank v_2, in the case of Rainbow. Such rank allows an attacker to isolate the central map polynomials from the first layer of Rainbow, and analogously recover the remaining layers with much lower efforts. In the context of Rainbow (BILLET; GILBERT, 2006; FAUGÈRE; LEVY-DIT-VEHEL; PERRET, 2008), its complexity was considered to be $q^{v_1+1} \cdot m \cdot (\frac{n^2}{2} - \frac{m^2}{6})$ field multiplications. However, recent developments, for example (BARDET et al., 2020, Tab. 3) and (NAKAMURA; WANG; IKEMATSU, 2020) have greatly improved the complexity of the attack. This fact is discussed briefly further into the text.

3.1.4.5 *HighRank attack (HR).*

Like MinRank, linear combinations of public key matrices are used to find the variables that are less likely to appear in the central map. Such a strategy is used to identify the last Rainbow layer and obtain the previous layers similarly. The complexity of the improved attack (DING et al., 2008) is $q^{o_u} \cdot \frac{n^3}{6}$ field multiplications.

3.1.4.6 *Rainbow-Band-Separation attack (RBS).*

An extension of the UOV reconciliation attack in the context of Rainbow (DING et al., 2008). It explores the fact that the central map matrix representation is composed of zeroes on its lower right corner. These yield quadratic equations that lead to an equivalent private key if solved. The complexity of this attack is given by the hardness of solving a polynomial system with n variables and $m+n-1$ equations (THOMAE, 2013, Thm. 3.30), which as seen above, is hard to estimate. It has been proposed by (PETZOLDT; BULYGIN; BUCHMANN, 2010c) that, in order to resist this attack, $n \geq \lceil \frac{5(m-1)}{3} \rceil$. Furthermore, recent works (PERLNER; SMITH-TONE, 2020; NAKAMURA et al., 2020) have shown improvements in the complexity of this attack, inducing a slight increase in parameter values submitted to the NIST standardization process. This fact is also discussed in the following.

3.1.4.7 *Side-channel attacks.*

We remark that none of our proposed Rainbow variants present constant time signature generation algorithms. It has been demonstrated that private keys can be recovered through correlation power analysis (PARK et al., 2018). A secure implementation is only given in the NIST standardization second-round submission of Rainbow (DING et al., 2019). Moreover, in Rainbow-η_2, a considerable amount of computation is added to the signature algorithm when one of the systems is not solvable. In a chosen message attack, one may observe the time spent on multiple signature generation steps and quickly check if the linear systems are solvable, thus obtaining information about the central map. Although there are no known attacks that make use of this technique, there may exist information leaks of this kind when applying our methods to Rainbow-like schemes.

3.1.4.8 Comparison.

In Table 1, we show the security level for several parameter sets in the literature and the attacks described above, except for side-channel attacks. We only show results for known secure parameter sets to prevent accidental endorsement of untested, and possibly insecure, instances. Further, we remark that our calculations are presented as approximate measures and reflect only an estimate of the complexities.

The Rainbow submission authors made conservative choices in the first (DING et al., 2017) and second (DING et al., 2019) rounds to fit security categories as requested by NIST. We apply our method to these recent proposals, and additionally choose parameters from (PETZOLDT, 2013, Tabs. 6.12, 9.8) for further comparison. The latter are named P-ℓ, where ℓ is the classical security level in bits given by the author.

Table 1 – Classical security level, in bits, for several instances of Rainbow.

Instance	Parameters	Direct	Collision	UOV	MR	HR
I-a	$(\mathbb{F}_{16}, 32, 32, 32)$	154	**128**	144	150	146
I-b	$(\mathbb{F}_{31}, 36, 28, 28)$	145	**139**	193	201	156
I-c	$(\mathbb{F}_{256}, 40, 24, 24)$	**138**	192	331	346	209
III-b	$(\mathbb{F}_{31}, 64, 32, 48)$	207	**199**	256	342	257
III-c	$(\mathbb{F}_{256}, 68, 36, 36)$	**206**	288	557	572	307
IV-a	$(\mathbb{F}_{16}, 56, 48, 48)$	231	**192**	243	248	212
V-c	$(\mathbb{F}_{256}, 92, 48, 48)$	**275**	384	751	765	405
VI-a	$(\mathbb{F}_{16}, 76, 64, 64)$	308	**256**	324	330	277
VI-b	$(\mathbb{F}_{31}, 84, 56, 56)$	289	**278**	435	442	298
P-080	$(\mathbb{F}_{256}, 17, 13, 13)$	**75**	104	143	159	118
P-100	$(\mathbb{F}_{256}, 26, 16, 17)$	**95**	132	209	232	152
P-128	$(\mathbb{F}_{256}, 36, 21, 22)$	**123**	172	290	313	193
P-192	$(\mathbb{F}_{256}, 63, 46, 22)$	**195**	272	706	532	**195**
P-256	$(\mathbb{F}_{256}, 85, 63, 30)$	266	372	956	709	**260**

Source – the author.

The minimum security level among different attacks for a parameter set is emphasized. In the case of direct attacks, we set $\omega = 2.4$ as the linear algebra constant, and simulate the complexity of solving an exactly determined polynomial system with m equations and variables. In the context of Rainbow, $n - m$ variables are fixed to attempt to solve the system. Since the complexity of this attack depends on the method used to find Gröbner bases, it is acceptable that there exists a small difference in the reported bit level, as in the case of P-ℓ. Moreover, the constraint of RBS is respected for all parameters sets with $q > 16$.

Recent developments of Rainbow cryptanalysis (DING et al., 2020) have improved the complexity of RBS and MinRank attacks. We present the results in Table 2 as given by the authors of the works mentioned earlier. We refer to the revised parameter sets simply by their targeted NIST security levels and convert the security level from logic gate counts to field multiplications, as per (DING et al., 2019, p. 35). Further information on closed formulas can be found on (NAKAMURA et al., 2020, Eq. 19) and (NAKAMURA; WANG; IKEMATSU, 2020, Sec. 5).

Table 2 – New parameters of Rainbow according to (DING et al., 2020, Sec. 4) and related classical security levels, in bits.

Instance	Parameters	Direct	Collision	RBS	UOV	MR	HR
I	$(\mathbb{F}_{16}, 36, 32, 32)$	159	**138**	142	152	157	145
III	$(\mathbb{F}_{256}, 68, 32, 48)$	227	**200**	210	430	221	403
V	$(\mathbb{F}_{256}, 96, 36, 64)$	278	**265**	274	560	289	532

Source – the author.

We note that most attacks look for unique structures within the private key. While our methods indeed modify the private key representation, it is still present in its entirety on the public key composition, which is readily available to malicious entities and can be possibly used to forge signatures. Thus, we suggest that the right choice of parameters is made whenever our methods are applied, according to (PETZOLDT; BULYGIN; BUCHMANN, 2010c), to protect the scheme instance against these attacks. We do not discard the possibility that specialized attacks exist, particularly ones that take into account multiple signatures, due to our fixing of vinegar variables.

3.2 STATISTICAL ARGUMENT

In this section, we argue that signatures generated by our proposal are comparably random to conventional Rainbow signatures. In Subsection 3.2.1, we look into the probability of matrices with elements in finite fields being invertible. In Subsection 3.2.2, we present a structural analysis of signatures created by our method.

3.2.1 Invertibility of the central map

Recall that, to create a Rainbow signature, a preimage of $\mathcal{C}$ needs to be calculated. Random guessing of vinegar variables is performed in order to create a solvable linear system. It is also known that the central map is expressed as multivariate quadratic polynomial systems, which can be themselves interpreted as multidimensional matrices of coefficients. With this in mind, we first derive the probability that a random matrix with elements in $\mathbb{F}$ is invertible.

Proposition 3.2.1 ((MULLEN; PANARIO, 2013, Remark 13.2.14)). *Given a square matrix M of order $n \times n$ such that $m_{ij} \in \mathbb{F}_q, 1 \leq i, j \leq n$, the probability that $M \in \mathsf{GL}_n(\mathbb{F}_q)$ is given by*

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

Proof. We recall that M is invertible if and only if $M \in \mathsf{GL}_n(\mathbb{F}_q)$. This happens if the set of all rows $m_i \in \mathbb{F}^n$ is linearly independent. The zero vector $(0,\dots,0) \in \mathbb{F}^n$ is linearly dependent of all other vectors. Thus, $m_1 \neq (0,\dots,0)$, with all other $q^n - 1$ possible vectors eligible. The row m_2 must not be any of the q multiples of m_1, and $q^n - q$ vectors remain. Without loss of generality, m_k must not be any of the q multiples of m_{k-1}, and so $q^n - q^{k-1}$ vectors can be selected. Then, the probability that all vectors chosen are linearly independent is

$$\Pi(q,n) = \prod_{k=1}^{n} \frac{q^n - q^{k-1}}{q^n} \quad = \prod_{k=1}^{n}(1-q^{-k}). \tag{3.14}$$

□

In the context of Rainbow, the number of layers directly influences $\Pi(q,n)$, since it dictates how many linear systems have to be solved. In other words, all square matrices of size $v_i, 1 \le i \le u$ need to be invertible to achieve a preimage under $\mathcal{C}$. Thus, the probability

$$\Pi(q,n,u) = \prod_{i=1}^{u}\prod_{k=1}^{v_{i+1}}(1-q^{-k}) \tag{3.15}$$

more accurately represents the upper bound for these chances. We denote tbe common case of $u = 2$ as $\Pi(q,n,2) = \widetilde{\Pi}(q,n)$, and we observe that $\Pi(q,n,1) = \Pi(q,n)$. Hence, schemes with more layers have a lower probability of success in the signature generation preimage step.

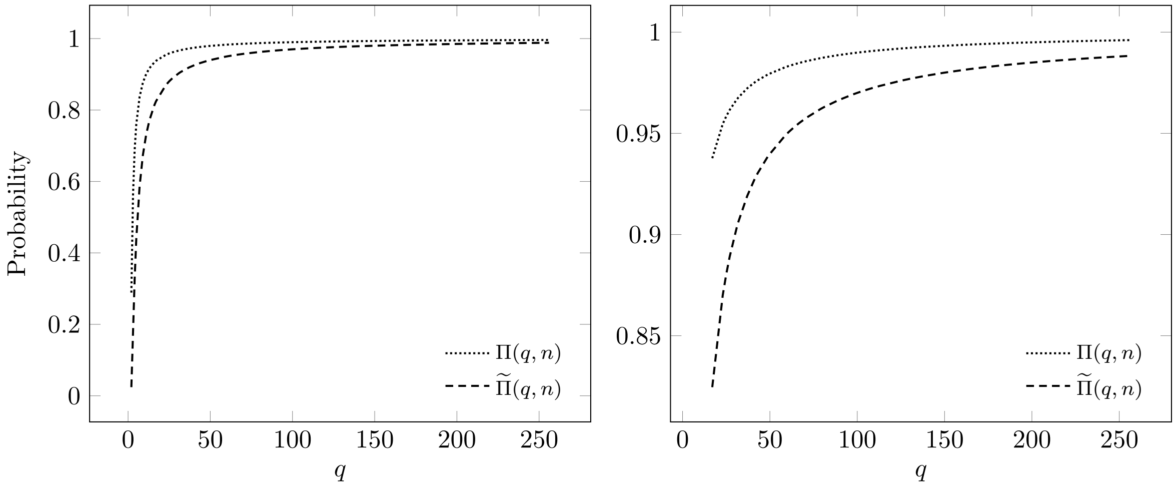
Parameters for Rainbow are selected according to a number of restrictions, imposed by attacks that may harm the security of the scheme. Furthermore, we recall Definition 2.2.9, which states that the central map can be represented as square matrices of order $n \times n$. Hence, we choose $56 \le n \le 90$ according to (PETZOLDT, 2013, Tabs. 6.4, 6.8, 6.13) and calculate the probability that a random matrix is invertible in finite fields of typical orders. Figure 2 depicts the lowest probabilities computed for the appropriate range. For instance, $\Pi(16,90) \approx 93.3594\%$ and $\Pi(256,56) \approx 99.6078\%$. To simulate layering, we set $v_i = i \cdot \lceil \frac{n}{u} \rceil$ and approximate to n when needed.

It is also useful to calculate $\Pi(q) = \lim_{n\to\infty}\Pi(q,n)$ to observe changes in the probability with the growth of the number of equations. We note that this limit has a well-known series representation.

Theorem 3.2.2 ((APOSTOL, 2010, Thm. 14.3))**.** *Euler's pentagonal number theorem states that*

$$\prod_{k=1}^{\infty}(1-q^k) = \sum_{k=-\infty}^{\infty}(-1)^k q^{\frac{3k^2+k}{2}}. \tag{3.16}$$

By simply swapping signs in the exponents, we can use this identity and obtain a fast approximation of the probability when n tends to infinity. We use the SageMath language, which offers arbitrary precision real numbers, to obtain these values and find out that, when



(a) $\operatorname{argmin}_{56 \leq n \leq 90}$ of $\Pi(q,n)$ and $\widetilde{\Pi}(q,n)$.

(b) Figure 2a, with $q \geq 16$.

Figure 2 – Probability of obtaining $M \in \mathsf{GL}_n(\mathbb{F}_q)$ with $2 \leq q \leq 256$ and q a prime power, given the quantity of layers of Rainbow.

Source – the author.

$n \geq 56$, $\Pi(q) \approx_{10^{-18}} \Pi(q,n)$ and $\widetilde{\Pi}(q) \approx_{10^{-8}} \widetilde{\Pi}(q,n)$. Thus, Figure 2 also accurately reflects the behaviour of $\Pi(q)$, that is, current values of n already reach effective upper bounds for this probability.

Hence, we note that the inversion event happens almost certainly. Moreover, this evidence shows that computing a preimage to sign a message happens at the first try with high probability in a wide range of Rainbow configurations. Therefore, we argue that the cost of generating signatures with Rainbow-η does not increase due to infrequent central map reconfiguration events.

3.2.2 Similarity of multiple signatures

Vinegar variables chosen to compute a preimage of the central map are an integral part of the preimage $\mathcal{C}(\mathbf{x}) = \mathbf{y}$. For instance, in the case $u = 2$, these make roughly a third of the output, considering common parameters for Rainbow. Further, we recall that there are approximately q^m possibilities for $\mathbf{x}$, for any $\mathbb{F}_q$. Our proposal eliminates this choice by locking vinegar variables into the private key. Hence, it is essential to know if such variables create patterns in which private information may leak through a multi-target attack. Throughout this subsection, we use the SageMath PRNG, which implements a front-end to the `/dev/urandom` Linux kernel space generator.

We recall that a message digest $\mathbf{h}$ is signed instead of the entire document. A secure cryptographic hash function shall produce an output that appears to be random. The application $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$ does not affect this behaviour, since the map is also random. Hence, we need not simulate this calculation in our analysis. According to Subsection 3.2.1, the operation $\mathcal{C}(\mathbf{x}) = \mathbf{y}$

creates a valid preimage with overwhelming probability, where the first v_1 elements of any $\mathbf{x}$ are the same.

We observe the distribution of field elements in vectors after the final function application, that is, $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$. Consider a t-tuple of signatures from an unmodified instance of Rainbow acting as a control group, that is, $Z'_t = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$, $t \in \mathbb{N}$. Let its "unpacked notation" be the $(n \times t)$-tuple $Z_t = (z_1^1, z_1^2, \ldots, z_1^n, z_2^1, \ldots, z_t^{n-1}, z_t^n)$ of elements of $\mathbb{F}_q$. When the first v_1 elements of the vector $\mathbf{x}$ are fixed due to the usage of Rainbow-η, we instead denote these by $\widetilde{Z}'_t$ and $\widetilde{Z}_t$.

We hypothesize that Z_t and $\widetilde{Z}_t$ behave similarly to observations sampled from a discrete uniform distribution, the random process used in choosing parts of a private key. We recall that any element of $\mathbb{F}_q$ is equally likely to appear in the matrix representation of a linear transformation or as a coefficient in an Oil–Vinegar polynomial.

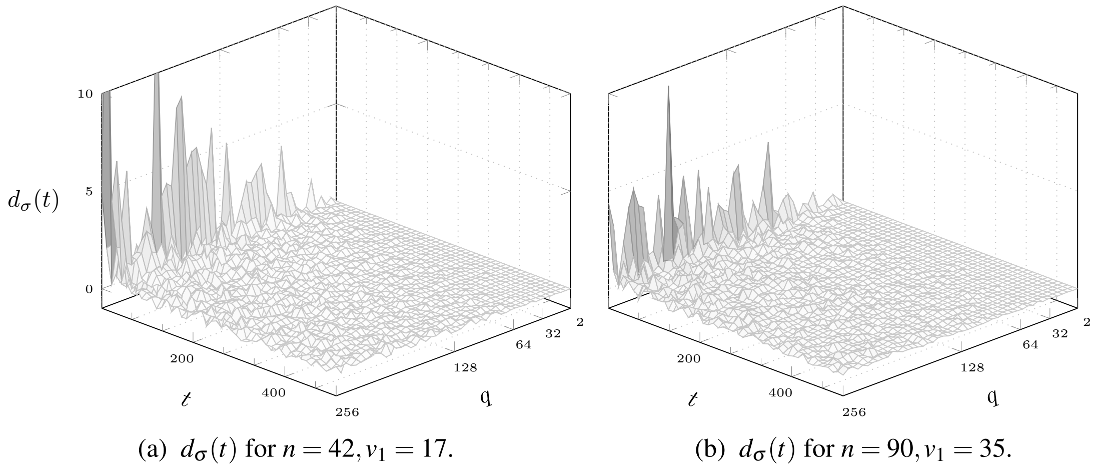
Definition 3.2.3. Let $\mathcal{U}(a,b)$ be a discrete uniform distribution with $a, b \in \mathbb{Z}$, $a \leq b$. The *standard deviation of $\mathcal{U}(a,b)$* is

$$\sigma = \sqrt{\frac{(b-a+1)^2 - 1}{12}}. \tag{3.17}$$

Let $\mathcal{U}(1,q)$ be the discrete uniform distribution that represents the chance of randomly picking any element from $\mathbb{F}_q$. We note that if the field characteristic is even, there exists a mapping to the natural numbers that allows this choice to be performed. Thus, we expect that

$$\lim_{t \to \infty} \sigma(\widetilde{Z}_t) = \sqrt{\frac{q^2 - 1}{12}}, \tag{3.18}$$

suggesting that greater values of n and t approximate faster to the theoretical standard deviation.



(a) $d_\sigma(t)$ for $n = 42, v_1 = 17$.

(b) $d_\sigma(t)$ for $n = 90, v_1 = 35$.

Figure 3 – Difference of standard deviations when $1 \leq t \leq 512$, and $2 \leq q \leq 256$ with q as a prime power.

Source – the author.

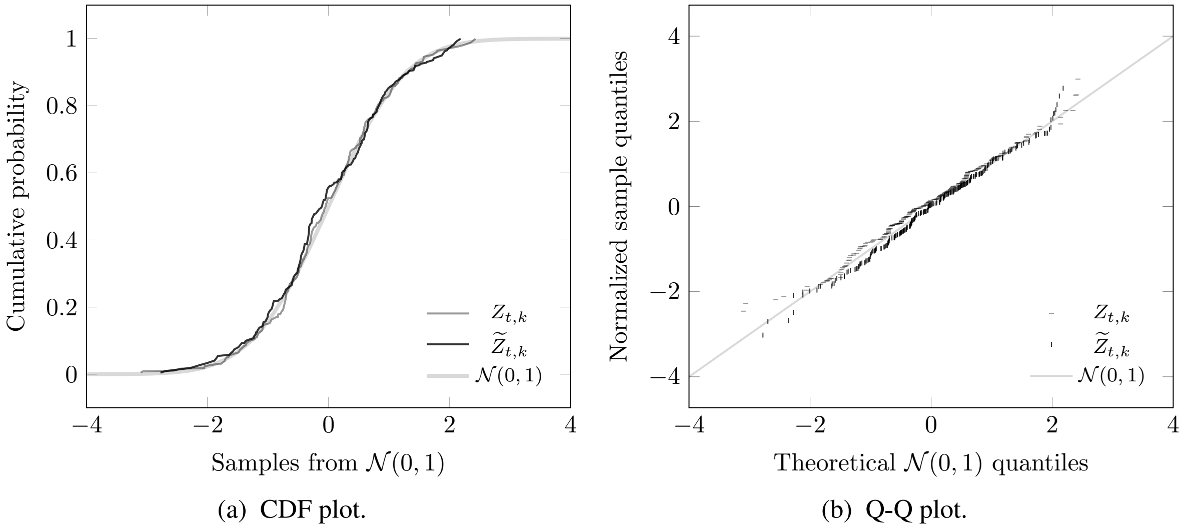
Let us denote the absolute difference between standard deviations for t as $d_\sigma(t) = |\sigma(Z_t) - \sigma(\widetilde{Z}_t)|$. Figure 3 shows the amplitude of such values for various values of q and t.

We note that the largest values of $d_\sigma(t)$ occur for finite fields of higher orders and lower t. For instance, given the vector space $\mathbb{F}_{223}^{42}$, we have $d_\sigma(1) \approx 8.25$, and for a slightly higher t, we obtain a much lower value $d_\sigma(11) \approx 0.24$. This behaviour is also observed within absolute differences of means, defined analogously as $d_\mu(t)$. The vector space $\mathbb{F}_{191}^{42}$ gives the values $d_\mu(1) \approx 5.31$ and, comparatively, $d_\mu(1) \approx 1.14$.

The comparison of expected with obtained standard deviations and means in our experiments gives positive results and confirms the law of large numbers. Still, it is interesting to look at the diffusion of values within $\widetilde{Z}_t$ and infer that it does not simply simulate the mean and standard deviation for a known discrete uniform. We count the amount of values for each class $1 \leq k \leq q$ and refer to them by $Z_{t,k}$ and $\widetilde{Z}_{t,k}$. By the central limit theorem, these counts should be normally distributed.

Figure 4 shows the cumulative distribution function (CDF) plot and the Q-Q (quantile-quantile) plot for such samples. The expected CDF, and examples for Z_t and $\widetilde{Z}_t$, show that all values are fairly distributed, with small variations due to the random generation of field elements. However, we note that this is due to the low number of classes, that is, the order of the finite field, and experimentally confirm that such discrepancies are largely reduced with $q = 2^{10}$.

The Q-Q plot, or normal probability plot, further confirms this behaviour. It is built by pairing the normalized counts $Z_{t,k}$ with quantiles from the standard normal distribution. Such quantiles are the normal order statistics, or rankits (IPSEN; JERNE, 1944, p. 349), which are hereafter referred to as $K \in \mathbb{R}^q$. Generating precise values for K is non-trivial, and we use the method below to yield approximate values.



(a) CDF plot.

(b) Q-Q plot.

Figure 4 – Distribution of counts of elements in Z_t and $\widetilde{Z}_t$ such that $t = 2^{16}$ for $\mathbb{F}_{256}^{90}$.

Source – the author.

Let $r \in \mathbb{N}$ be a precision parameter, and $\min(S,i)$ represent the i-th smallest value in a non-empty set S with a partial order. A number $r \times q$ of samples are taken from a standard normal distribution $\mathcal{N}(0,1)$ and organized into q-tuples $K_1,\dots,K_r$. Then, K is composed of the means of the i-th smallest values in all K_j, that is,

$$K = \big(\mu(\{\min(K_j,i) : 1 \le j \le r\}) : 1 \le i \le q\big)\,. \tag{3.19}$$

We have found experimentally that the lowest precision parameter $r = 1$ is still useful to interpret the graph. The resulting plot shows that points are sufficiently close to the $y = x$ expected line.

Our argument indicates that, even if part of the preimage created by the central map is fixed, the remaining linear transformation disrupts this pattern with high efficacy. Hence, an attacker with possession of multiple signatures created by our method would not be more capable of forging a new signature or deducing private information through strict statistical methods, than an attacker with no such knowledge.

3.3 ENHANCEMENT OF EXISTING SCHEMES

Our method does not depend on structures inserted on the private key. Consequently, it can be applied to all known Rainbow-like schemes. We experiment with several sets of parameters and observe the reduction of private key sizes. It is known that there are various limitations for the choice of parameters that lead to secure instances of Rainbow (PETZOLDT; BULYGIN; BUCHMANN, 2010c). We implement several known guidelines and confirm that our proposal does indeed work for a broad range of parameters. Furthermore, we remark that comparisons are made considering the usage of homogeneous quadratic polynomials, and simplified linear transformations due to Theorem 2.3.7.

We show results for the application of our method in Table 3, considering parameter sets of Tables 1 and 2. Indeed, the choice of v_1 remarkably affects the results. Moreover, a minimal value of o_u is also known to reduce the private key size further. We suggest that $v_1 \ge o_u$ as much as possible to maximize the results of our method. However, we remark that one must set sufficient parameters for o_i such that the scheme still resists direct and UOV attacks.

In the case of Rainbow variants, schemes claim optimizations of the private key often through the inclusion of inner structuring. It is imperative to understand such structures to measure the impact of our method within the context of these schemes. For instance, it may be the case that a method introduces sparseness related to specific vinegar variables. Thus, the reduction would not be equally distributed over the private key elements, and our method would have its efficiency reduced.

To the best of our knowledge, it is not trivial to combine our strategy with the schemes presented in Subsection 1.3.1, and resulting gains would be reduced. Furthermore, several of those schemes were subsequently broken, or new parameters were proposed, suggesting that such a combination would not be productive. We thus consider only schemes that reduce the public key size, that is, CyclicRainbow (PETZOLDT; BULYGIN; BUCHMANN, 2010b) and RainbowLRS2 (PETZOLDT, 2013, Sec. 9.2).

Table 3 – Reduction of Rainbow key sizes, in bytes, for various instances of the scheme.

Instance	Parameters	n	m	#sk	#sk$^{\eta}$	Difference
I-a	$(\mathbb{F}_{16}, 32, 32, 32)$	96	64	92 928	26 896	−71.06%
I-b	$(\mathbb{F}_{31}, 36, 28, 28)$	92	56	105 916	24 628	−76.75%
I-c	$(\mathbb{F}_{256}, 40, 24, 24)$	88	48	132 576	24 136	−81.79%
III-b	$(\mathbb{F}_{31}, 64, 32, 48)$	144	80	389 975	71 552	−81.65%
III-c	$(\mathbb{F}_{256}, 68, 36, 36)$	140	72	511 416	78 188	−84.71%
IV-a	$(\mathbb{F}_{16}, 56, 48, 48)$	152	96	358 656	88 540	−75.31%
V-c	$(\mathbb{F}_{256}, 92, 48, 48)$	188	96	1 227 072	180 572	−85.28%
VI-a	$(\mathbb{F}_{16}, 76, 64, 64)$	204	128	860 800	206 630	−76.00%
VI-b	$(\mathbb{F}_{31}, 84, 56, 56)$	196	112	980 524	187 172	−80.91%
P-080	$(\mathbb{F}_{256}, 17, 13, 13)$	43	26	16 757	4177	−75.07%
P-100	$(\mathbb{F}_{256}, 26, 16, 17)$	59	33	41 163	8364	−79.68%
P-128	$(\mathbb{F}_{256}, 36, 21, 22)$	79	43	96 288	17 754	−81.56%
P-192	$(\mathbb{F}_{256}, 63, 46, 22)$	131	68	416 998	52 417	−87.43%
P-256	$(\mathbb{F}_{256}, 85, 63, 30)$	178	93	1 043 295	128 950	−87.64%
I	$(\mathbb{F}_{16}, 36, 32, 32)$	100	64	103 616	27 026	−73.92%
III	$(\mathbb{F}_{256}, 68, 32, 48)$	148	80	626 016	107 652	−82.80%
V	$(\mathbb{F}_{256}, 96, 36, 64)$	196	100	1 408 704	204 384	−85.49%

Source – the author.

We compare key pair sizes when our method is used alongside Rainbow variants that reduce the public key size. Table 4 presents results for some sets of parameters from Table 3. We calculate #pk for the variants according to (PETZOLDT, 2013, Eqs. 9.2, 9.4) and (PETZOLDT, 2020). As per (PETZOLDT, 2013, Remark 9.1), we note that $q = 16$ and $q = 31$ are not considered due to a security restriction of RainbowLRS2. The “nCyclic” method is a small improvement over CyclicRainbow featured on the second-round submission of Rainbow to the NIST standardization process (DING et al., 2019). We obtain positive results, with key pair size reductions of up to a factor of 3.

The use of CyclicRainbow or RainbowLRS2 with Rainbow-η_2 is recommended. These variants are based on the linear relations described in Subsection 3.1 and resulting implementations may be effortlessly modified to use our proposal. Moreover, if higher parameters are needed, e.g. a security level of 256 bits, we note that the key pair is reduced more aggressively. Thus, our results reflect changes over a wide variety of platforms and possible Rainbow deployments that benefit from lower storage requirements.

We also briefly discuss the effect of these changes on the signature generation step overall performance. In the case of Rainbow-η_1, it does not vary significantly due to the fast regeneration of the central map elements from a given PRNG and $\mathbf{S}$. On the other hand, Rainbow-η_2 uses elaborate techniques to reconstruct the central map if vinegar variables are not suitable.

Table 4 – Total reduction of Rainbow key pairs, in bytes, for variants of the scheme.

Instance	Variant	#sk	$\#\text{sk}^{\eta}$	#pk	Difference
P-080	Classic	16757	4177	24596	−30.42%
	Cyclic			9474	−66.99%
	LRS2			8645	−68.99%
P-100	Classic	41163	8364	58410	−32.94%
	Cyclic			20266	−71.25%
	LRS2			18682	−72.84%
P-128	Classic	96288	17754	135880	−33.83%
	Cyclic			44971	−72.98%
	LRS2			42107	−74.22%
I	Classic	103616	27026	161600	−28.88%
	nCyclic			60160	−67.13%
III	Classic	626016	107652	882080	−34.37%
	nCyclic			264576	−75.32%
V	Classic	1408704	204384	1930600	−36.07%
	nCyclic			536104	−77.83%

Source – the author.

This process is not without cost, and it may negatively affect the average signature generation time, as seen in (PETZOLDT, 2020). Still, by making use of Rainbow-η_3, these computations are entirely avoided by choosing a new salt instead of new vinegar variables, reducing the inherent overhead.

We conclude that Rainbow-η offers several competitive features and may act as a replacement for the original Rainbow proposal. For instance, signatures produced with Rainbow-η are statistically indistinguishable from signatures created with the unoptimized method, preventing related attacks. Moreover, our method allows a reasonable reduction in private key sizes and it is applicable to variants that optimize public keys.

4 VARIANT OF KIPNIS–SHAMIR ATTACK

As seen in Chapter 3, several arguments were made in order to present support for the security of our Rainbow-η proposal. We recall that the inner workings of the scheme are not modified, and a sideline modification is done instead, which preserves the randomness of signatures. As summarized in Section 1.3, there is encouraging evidence that adding further structure to the central map is dangerous to the security of Oil–Vinegar schemes.

We intentionally avoid this strategy with our proposal of Chapter 3. However, we show that it is not secure due to a leakage of information from the private key through signatures. In Section 4.1, we briefly describe the Kipnis–Shamir attack and present a variation of this attack that produces an equivalent key for a scheme that makes use of Rainbow-η. In Section 4.2, we show the practical consequences of this attack, and we argue that fixing less vinegar variables still leads to an efficient attack.

4.1 ATTACK DESCRIPTION

In this section, we show how to mount a key recovery attack against a scheme that uses the Rainbow-η strategy. In Subsection 4.1.1, we recall the Kipnis–Shamir attack, the first cryptanalytic method that specifically targeted Oil–Vinegar signature schemes. In Subsection 4.1.2, we present our version of the attack that works against the proposal of Chapter 3.

4.1.1 Original method

The Kipnis–Shamir attack, or UOV attack, was first given in (KIPNIS; SHAMIR, 1998) as proof of insecurity of the balanced Oil–Vinegar signature scheme. The fact that $v = o$ allows for the leakage of information in the public key that can be used to reconstruct an equivalent private key. This method led to the creation of UOV as a countermeasure (KIPNIS; PATARIN; GOUBIN, 1999) with $v = 2o$. The attack was generalized on the same work, but it is computationally hard to perform if the difference $v - o$ is not small enough.

The UOV attack stands relevant in the cryptanalysis of Oil–Vinegar signature schemes, as seen in Section 1.3. We describe the attack for $v = o$ and finite fields of odd characteristic, following the rationale of (DING; GOWER; SCHMIDT, 2006, Sec. 3.2). We refer the reader to (CAO et al., 2011) for the case of even characteristic and (KIPNIS; PATARIN; GOUBIN, 1999) for the case of $v > o$. Without loss of generality, we restrict the polynomials of $\mathcal{C}$ and $\mathcal{P}$ to be homogeneous quadratic.

Consider the vector space $\mathbb{F}^o$ that denotes all possible messages, and the vector space $\mathbb{F}^n$ that denotes all possible signatures from a balanced Oil–Vinegar signature scheme, with $n = v + o$. Its private and public keys are given in Equations 2.14 and 2.15, respectively. We introduce further vector spaces that are essential to discuss the attack.

Definition 4.1.1. Consider the subset of $\mathbb{F}^n$ composed of all vectors with the first v coordinates equal to zero, that is, $\{(0,\dots,0,x_{v+1},\dots,x_n) \in \mathbb{F}^n\}$. The vector subspace of $\mathbb{F}^n$ formed by this set with the usual binary operations is the *oil subspace*, and denoted by $\mathcal{O}$. The preimage of $\mathcal{O}$ under $\mathcal{L}_2$ is given by $\widetilde{\mathcal{O}}$.

Definition 4.1.2. Consider the subset of $\mathbb{F}^n$ composed of all vectors with the last o coordinates equal to zero, that is, $\{(x_1,\dots,x_v,0,\dots,0) \in \mathbb{F}^n\}$. The vector subspace of $\mathbb{F}^n$ formed by this set with the usual binary operations is the *vinegar subspace*, and denoted by $\mathcal{V}$. The preimage of $\mathcal{V}$ under $\mathcal{L}_2$ is given by $\widetilde{\mathcal{V}}$.

The main observation of the attack is that combinations of matrices associated with the polynomial systems of $\mathcal{C}$ by Definition 2.3.3 are precisely structured. Since the public key preserves this arrangement, it can be used by the attacker to forge signatures. A more precise characterization of this fact is given below.

Lemma 4.1.3 ((KIPNIS; PATARIN; GOUBIN, 1999, Lemma 1))**.** Given any polynomials $f^{(i)}, f^{(j)} \in \mathsf{OVH}(v,o)$, let $\phi^{(i)} : \mathbb{F}^n \to \mathbb{F}^n$ and $\phi^{(j)} : \mathbb{F}^n \to \mathbb{F}^n$ be the linear transformations associated with the matrices $\widetilde{M}(f^{(i)})$ and $\widetilde{M}(f^{(j)})$ of Definition 2.3.3, respectively. If $\widetilde{M}(f^{(j)})$ is invertible, then so is $\phi^{(j)}$ and thus $\mathcal{O}$ is an invariant subspace of $\phi^{(i)} \circ (\phi^{(j)})^{-1}$.

We recall that $f^{(i)}$ and $f^{(j)}$ are part of an Oil–Vinegar central map. Informally, any symmetric matrix representation of a polynomial of $\mathcal{C}$ maps $\mathcal{O}$ to $\mathcal{V}$ due to its Oil–Vinegar structure. The inverse of this operation gives a mapping from $\mathcal{V}$ to $\mathcal{O}$. By composing the linear transformations obtained from matrices of two distinct polynomials as described above, the oil space becomes an invariant subspace with regards to this new linear transformation.

Further, we have to account for $\mathcal{L}_2$, since it is composed with $\mathcal{C}$ to form the public key. As argued in (KIPNIS; SHAMIR, 1998, p. 261), the aforementioned behaviour is analogous for any combination of matrices associated with polynomials of $\mathcal{P}$. Such a matrix maps $\widetilde{\mathcal{O}}$ to $\widetilde{\mathcal{V}}$, and its inverse maps $\widetilde{\mathcal{V}}$ to $\widetilde{\mathcal{O}}$. With several distinct combinations of the aforementioned form, it is possible to obtain the *common* invariant subspace $\widetilde{\mathcal{O}}$ (KIPNIS; PATARIN; GOUBIN, 1999, Thm. 1).

If the vector subspace $\widetilde{\mathcal{O}}$ is known, it is straightforward to obtain its basis B with dimension o. Consider $\psi(\widetilde{\mathcal{L}_2}) \in \mathsf{GL}_n(\mathbb{F})$ as the matrix $\mathbf{I}_n$ with elements of B as its last o columns. This matrix has the shape of ω in Theorem 2.3.4. Consequently, the associated linear transformation $\widetilde{\mathcal{L}_2}$ allows for the construction of an equivalent key of the form

$$\mathsf{sk} = (\mathcal{P} \circ \widetilde{\mathcal{L}_2}, \widetilde{\mathcal{L}_2}^{-1}). \tag{4.1}$$

This can be verified by inspection of the associated symmetric matrix representations, as given in (DING; GOWER; SCHMIDT, 2006, p. 71).

The computational complexity of the general attack is $q^{n-1-2\cdot o_u} \cdot o_u^4$ field multiplications. Indeed, the attack is performed in polynomial time in the case of the balanced Oil and

Vinegar scheme, which has $n = 2v = 2o$ and $u = 1$. The UOV attack is also applicable to instances of Rainbow. Consider a Rainbow private key as shown in Equation (2.24). It is evident that the function composition $\mathcal{D} = \mathcal{L}_1 \circ \mathcal{C}$ preserves the Oil–Vinegar trapdoor, since $\mathcal{L}_1$ only permutes the coefficients of the polynomials in the central map. The resulting map $\mathcal{D}$ is an Oil–Vinegar map with $v_u = v_1 + o_1 + \cdots + o_{u-1}$ vinegar variables and o_u oil variables, but with extra $m - o_1$ polynomials. Such polynomials play a factor in the complexity of our attack, as seen below.

4.1.2 Our attack

We recall that our proposal Rainbow-η, as described in Section 3.1, is a modification based on fixing vinegar variables early in the key generation step. By pre-substituting such variables in the central map, it can be stored more efficiently. Further, we ensure that any message can be signed through a reconstruction of the central map if its associated preimage calculation does not succeed. Moreover, we impose no limitation on parameters such that their secure selection is always encouraged.

Nonetheless, we identify that a sequence of signatures created with the same vinegar variables can be used to mount a key recovery attack. Then, a malicious entity can create signature forgeries in polynomial time, similar to the UOV attack. The critical observation is that signatures produced with Rainbow-η are used to obtain vectors of $\widetilde{\mathcal{O}}$. Consequently, an equivalent linear transformation $\widetilde{\mathcal{L}}_2$ can be recovered and composed with $\mathcal{P}$ to yield an equivalent private key.

The authors of (SHIM; KOO, 2020) present a similar attack, in which the threat model assumes that vinegar values are fixed by inducing faults through physical sources or software. They show that, in fact, any fixed vinegar variables are enough to theoretically break Rainbow (SHIM; KOO, 2020, Thm. 4), i.e. a generalized Rainbow-η is also insecure. We further explore this in Subsection 4.2.2. We note that it is difficult to induce faults in vinegar variables for even a single signature generation procedure (MUS; ISLAM; SUNAR, 2020). Therefore, we emphasize the importance of randomized vinegar variables in the Rainbow signature scheme by using our independently discovered attack to break our proposal of Chapter 3.

Without loss of generality, we describe the attack for UOV, that is, a Rainbow scheme with a single layer and $\mathcal{L}_1 = \mathsf{id}$. However, we note that this strategy can also be applied to a Rainbow scheme, given that it can be interpreted as a UOV scheme, as previously discussed. Consider any instance $\mathsf{UOV}(\mathbb{F}, v, o)$ as per Subsection 2.3.1. We recall that $n = v + o$ is the number of variables, and $m = o$ is the number of equations in $\mathcal{C}$ and $\mathcal{P}$.

An attacker must collect $m + 1$ signatures known to have been produced with the same vinegar variables through Rainbow-η for the attack to work. Let the set of collected signatures be denoted as $\rho = \{\sigma_1, \ldots, \sigma_{m+1}\}$. Any preimage of $\mathcal{C}$ always starts with $\widetilde{V}_1 = (x_1, \ldots, x_v)$ in the case of Rainbow-η. Thus, by inverting the linear transformation used to scramble the preimage, we obtain $\mathcal{L}_2(\sigma_i) = (x_1, \ldots, x_v, *, \ldots, *)$. Then, for any $1 \leq i, j \leq m+1$ with $i \neq j$, we note that

$$\mathcal{L}_2(\sigma_i - \sigma_j) = (0, \ldots, 0, *, \ldots, *).$$

By Definition 4.1.1, any subtraction among signatures of the above form is in $\widetilde{\mathcal{O}}$. This is an effective characterization of the target subspace in the UOV attack. The only remaining task is to build a basis B of $\widetilde{\mathcal{O}}$, which has dimension m. We first show that $m+1$ is the minimal amount of vectors needed to build B.

Proposition 4.1.4. *Consider a set R of $m+1$ linearly independent vectors in $\mathbb{F}^n$. The maximum cardinality of a set $\widetilde{R}$ composed of linearly independent subtractions between any pair of vectors in R, such that all vectors are used, is m.*

Proof. We model the problem as a graph instance. Let $G = (V, E)$ be an undirected connected graph such that $V = R$ and $E = \widetilde{R}$ using the mapping $f : V \times V \to \widetilde{R}$, such that for any $\mathbf{v_i}, \mathbf{v_j} \in V$, then $f((\mathbf{v_i}, \mathbf{v_j})) = \mathbf{v_i} - \mathbf{v_j}$.

We first show that one can obtain m linearly independent subtractions from vectors in R. For instance, consider the star graph. Without loss of generality, we choose $\mathbf{v_1}$ as the vertex with maximal degree. The resulting graph is composed of edges $f((\mathbf{v_1}, \mathbf{v_i}))$ for $2 \leq i \leq m+1$. Any linear combination of such subtractions is linearly independent, since

$$\begin{aligned} c_2(\mathbf{v_1} - \mathbf{v_2}) + c_3(\mathbf{v_1} - \mathbf{v_3}) + \cdots + c_{m+1}(\mathbf{v_1} - \mathbf{v_{m+1}}) = \\ (c_2 + \cdots + c_{m+1})\mathbf{v_1} - c_2\mathbf{v_2} - \cdots - c_{m+1}\mathbf{v_{m+1}} = 0 \end{aligned}$$

implies that $c_2 = \cdots = c_{m+1} = 0$, using the fact that all vectors in V are linearly independent. In other words, the resulting graph is a tree, since it is connected and has exactly $|V| - 1$ edges.

Now, we argue that a set of linearly dependent subtractions of vectors in V is obtained if one extra edge is added to the graph. Since G is a tree, we obtain a cycle by further adjoining any two vertices. For instance, in the case of a star graph, the coefficients of $\mathbf{v_1} - \mathbf{v_i}$ for all i not in the cycle are zero. For the remaining three coefficients, a combination of $\{1, -1\}$ is needed. □

We note that if an attacker has possession of a larger number of signatures produced with the same vinegar variables, the resulting graph is not necessarily connected. However, a spanning tree of the first $m+1$ signatures can always be constructed and the remaining ones discarded. Moreover, if we consider signatures to be labelled vertices, there are at most $(m+1)^{(m+1)-2}$ spanning trees that respect the graph model of Proposition 4.1.4, by Cayley's formula. This is an upper bound on the number of possible choices for $\widetilde{R}$ with maximal cardinality, since not necessarily all edges are themselves linearly independent.

Let $\widetilde{\rho} = \{\chi_i : 1 \leq i \leq m\}$ be a set of vectors built from subtractions among signatures of ρ. A naive strategy is $\widetilde{\rho} = \{\sigma_1 - \sigma_{i+1} \mid 1 \leq i \leq m\}$, which forms a star graph, considering the approach of Proposition 4.1.4. On the other hand, $\widetilde{\rho} = \{\sigma_i - \sigma_{i+1} \mid 1 \leq i \leq m\}$ forms a tree with no branches, but with no functional difference. Furthermore, as seen in Subsection 3.2.2, signatures are statistically similar to random elements of the underlying vector space. Thus, it is expected with high probability that all χ_i are immediately linearly independent.

Given such a basis B, we proceed with the last part of the UOV attack as described previously. Let $\chi_i^{(j)}$ be the j-th element of the i-th vector in $\widetilde{\rho}$, with $1 \leq i \leq m$ and $1 \leq j \leq n$. Then, we have that

$$\psi(\widetilde{\mathcal{L}_2}) = \begin{bmatrix} 1 & 0 & 0 & \cdots & \chi_1^{(1)} & \cdots & \chi_m^{(1)} \\ 0 & 1 & 0 & \cdots & \chi_1^{(2)} & \cdots & \chi_m^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \chi_1^{(n)} & \cdots & \chi_m^{(n)} \end{bmatrix}. \tag{4.2}$$

We obtain the associated linear transformation $\widetilde{\mathcal{L}_2}$ and proceed to recover an equivalent key as per Equation (4.1). By using this method, an attacker can create sk' and forge signatures for any message. In fact, if the method of Rainbow-η is not applied to the equivalent key, the forged signatures do not leak any information.

In the case of $u > 1$, the rationale above is maintained since we are able to interpret an instance of Rainbow as a larger UOV scheme. However, we are able to fix only the first v_1 variables in preimages of $\mathcal{C}$. Then, a vector χ_i is not in $\widetilde{\mathcal{O}}$ with overwhelming probability, and thus $\psi(\widetilde{\mathcal{L}_2})$ is incomplete. Hence, we need to obtain the value of the remaining $x_{v_1+1}, \ldots, x_{v_u}$ variables for a number m of signatures. This is equivalent to solving a polynomial system in $\mathsf{MQ}(m - o_u, m)$. The Hybrid approach (BETTALE; FAUGÈRE; PERRET, 2012) is an algorithm used to solve such instances of overdetermined polynomial systems, with an estimate asymptotic complexity of $2^{n \cdot (3.31 - 3.62 \log_2(q)^{-1})}$ field operations. Thus, the attack is not carried in polynomial time, but the effort needed to forge a signature is greatly reduced if Rainbow-η is used with $u = 2$ or greater.

Interestingly, this attack can be used to “recover” a lost private key of a legitimate signer. It must previously compute m linear independent preimages of $\mathcal{C}$ with the same vinegar variables and store these securely. If the private key is lost, the signer would only need to fetch its public key and compute an equivalent private key by the method given above.

4.2 MOUNTING THE ATTACK IN PRACTICE

In this section, we show the efforts associated with performing our attack, and a possible strategy to repair our proposal, with a clear trade-off between private key sizes and security levels. In Subsection 4.2.1, we discuss the probabilities of applying the attack to vulnerable Rainbow-like schemes in practical settings. In Subsection 4.2.2, we examine the effects of fixing a fraction of all vinegar variables with regards to attack probabilities.

4.2.1 Collection of signatures

As discussed in Subsection 4.1.2, the collection of signatures produced with Rainbow-η is pivotal to carry on with our attack. Thus, we show the probabilities associated with picking a valid subset of items from a large number of amassed signatures to, in turn, obtain an equivalent key. Due to its large keys, Rainbow is more likely to be used in situations of rare key transmission, e.g. signing software packages. Thus, a real-life analogy would be a software repository with several maintainers that sign their own packages, and at least one chooses to use Rainbow-η.

We remark that there is no way to identify if a signature is produced with Rainbow-η except by observation of the public implementation. However, it is simple to modify the key and signature generation algorithms to yield either conventional or reduced private keys by choice of the signer. If an attacker has no access to the signing device, it is impossible to discern whether the Rainbow-η optimization is used. Otherwise, side-channel attacks like (KRÄMER; LOIERO, 2019) and (SHIM; PARK; BAEK, 2015) may be used to increase chances that "weak" signatures are produced.

Let us consider that a signer produces a number $N \in \mathbb{N}$ of signatures such that $K \leq N$ of those are created with the same vinegar variables. Without loss of generality, let $K = \lceil \alpha N \rceil$ with $0 \leq \alpha \leq 1$. An attacker increases the odds of obtaining useful signatures for the attack if a target is singled out. Nonetheless, we observe such probabilities for various values of α and model this behaviour as follows. Given N and K as mentioned above, we wish to obtain $k \leq K$ signatures that were produced with Rainbow-η from a sample size $n \leq N$.

By the description of our attack in Subsection 4.1.2, the attacker needs k distinct signatures, or in other words, sampling is *without replacement*. The probability distribution that describes this situation is given below.

Definition 4.2.1. Given $N, K, n, k \in \mathbb{N}$ with $n \leq N$, $K \leq N$ and $0 \leq k \leq K$, if the probability mass function (PMF) of a discrete random variable $\mathbf{X}$ is

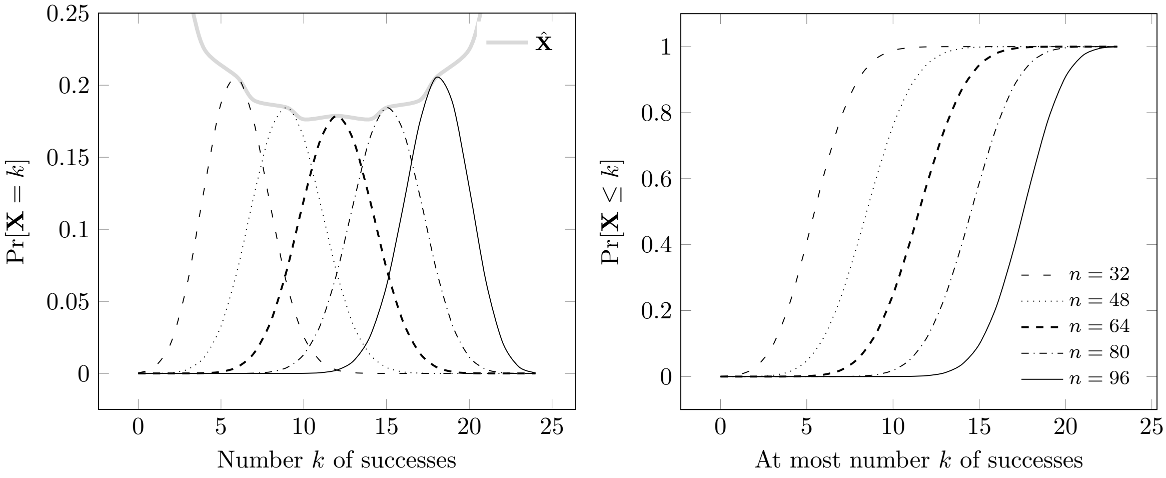
$$\Pr[\mathbf{X} = k] = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}},$$

then $\mathbf{X}$ follows the *hypergeometric distribution*. The peak of this PMF, or its *mode*, is

$$\widehat{\mathbf{X}} = \left\lfloor \frac{(n+1)(K+1)}{N+2} \right\rfloor .$$

The probability of sampling a valid subset of signatures is then equivalent to calculating $1 - \Pr[\mathbf{X} \leq k]$, which is the description of the *complementary* cumulative distribution function (CDF).

We now explore the probability that, given n draws from a population of size N with K "weak" signatures, there exist at least k successes. Figure 5 shows the PMF to the left, and the CDF to the right, of a sample hypergeometric distribution. For instance, we note that for a factor $\alpha = \frac{1}{5}$ and $N = 128$, if an attacker performs sampling without replacement of $\frac{1}{4}$ of the population, that is $n = 32$, there exists a chance of $\approx 1\%$ that $k = 10$ or more signatures with the desired feature are obtained. On the other hand, if $\frac{1}{2}$ of the population is sampled, or $n = 64$, the probability is $\approx 75\%$. Indeed, the probability of obtaining a higher number of successes grows with the number of draws.



(a) PMF plot.

(b) CDF plot.

Figure 5 – Hypergeometric distribution with $N = 128$, $K = 24$ and $n \in \{32, 48, 64, 80, 96\}$.

Source – the author.

However, the sample size is easily controlled by the attacker, while the α factor may be unknown. To this end, a remarkable fact about the hypergeometric distribution is that it is *symmetric* on the K and n parameters. Such symmetry is verified by expanding the PMF definition into binomials for both cases and checking for the desired equality. Thus, Figure 5 also represents the behaviour of various α factors in the form of different quantities of signatures eligible for the attack, that is, $K \in \{32, 48, 64, 80, 96\}$ and a fixed number of samples $n = 24$.

It is thus expected that a large enough number of signatures is collected for a reasonable chance of obtaining a valid sample. For example, given $N = 2^{12}$ and $K = 2^{10}$, or $\alpha = \frac{1}{4}$, and $n = 2^8$, then $\widehat{\mathbf{X}} = 64$, which is equal to the number of equations m of a Rainbow instance with minimal security level (DING et al., 2019, Sec. 2.8). The attack description in Subsection 4.1.2 requires $m+1$ "weak" signatures as a parameter. Thus, the associated probability of obtaining at least 65 successes is $\Pr[\mathbf{X} > \widehat{\mathbf{X}}] \approx 46.6\%$.

The identification problem remains non-trivial if n is much larger than k, since the attacker still does not know which sampled signatures are useful to perform the attack. Ideally, a sample should return exactly the desired number of "weak" signatures, that is, $n = k$. In this case, the PMF is given by

$$\frac{\binom{K}{k}}{\binom{N}{k}} = \frac{K!(N-k)!}{N!(K-k)!},$$

which yields small probabilities even for modest parameters. For example, given $N = 2^8$ and $K = 2^7$, or $\alpha = \frac{1}{2}$, and $n = k = 2^6$, we get $\Pr[\mathbf{X} = k] \approx 1.2577 \times 10^{-24}$. If $n = k = 2^7$, we have $\Pr[\mathbf{X} = k] \approx 1.7335 \times 10^{-76}$. Hence, on average, it is not trivial to collect signatures and attack a Rainbow instance with adequate security level if α is sufficiently large.

We infer that the success of an attacker is directly proportional to the maximization of the number of draws or the α factor. Obtaining a collection of signatures of which a subset is useful to perform the attack is trivial, but the effort associated with locating such subset is meaningful. Still, by bounding the security level of the resulting scheme to the chances associated with a best-case scenario of the probabilities previously discussed, we obtain an efficient, polynomial-time procedure.

4.2.2 Choice of vinegar threshold

It is clear from Subsection 4.2.1 that some effort is needed to apply the attack of Subsection 4.1.2 in practice. However, such a mathematical method presents a definite advantage for an attacker to obtain an equivalent key from a well-targeted entity. As previously mentioned, it is known that any fixed vinegar variables are a thread to the security of Rainbow. To properly discuss this fact, we first give a general description of our original proposal, based on a configuration parameter that controls how vinegar variables are fixed. Moreover, we discuss the probabilities of performing our attack if some level of randomness is maintained in the central map preimage.

We recall that $\widetilde{V}_1$ is the sequence of vinegar variables used on the signature generation algorithm of Rainbow-η. In the original scheme, it is chosen randomly for every signing process, yielding roughly q^{v_1} possible preimages for a message digest with q as the order of the underlying field. In the case of Rainbow-η, there is only one preimage if the partial evaluation of the central map gives an exactly determined polynomial system. We recall a well-known concept that allows the quantification of such preimages and other features of our new proposal.

Definition 4.2.2. Given $k \in \mathbb{N}$ and a binary vector $\mathbf{x} \in \mathbb{F}_2^k$, the number of non-zero elements of $\mathbf{x}$, or

$$\mathrm{wt}(\mathbf{x}) = |\{i \mid x_i \neq 0, 1 \leq i \leq k\}|, \tag{4.3}$$

is the *Hamming weight of* $\mathbf{x}$.

Let $\mathbf{v} \in \mathbb{F}_2^{v_1}$ be the *vinegar threshold*, a sequence of bits that encodes which elements from $\widetilde{V}_1$ are to be fixed. More precisely, for $1 \leq i \leq v_1$, if $\mathbf{v}_i = 0$ then the i-th element of $\widetilde{V}_1$ is swapped with a randomly chosen element from $\mathbb{F}_q$, and if $\mathbf{v}_i = 1$, it remains unchanged. Evidently, if $\mathbf{v} = (0, \ldots, 0)$, we turn back to the original Rainbow scheme, and for $\mathbf{v} = (1, \ldots, 1)$, we get the schemes of Section 3.1. Therefore, the number of possible signatures for a given message digest is $\approx q^{v_1 - \mathrm{wt}(\mathbf{v})}$. Let us denote this method as Rainbow-η_4. We remark that it is compatible with any of the schemes of Section 3.1.

4.2.2.1 Key generation.

Generate a $\mathbf{v} \xleftarrow{r} \mathbb{F}_2^{v_1}$ such that $\mathrm{wt}(\mathbf{v}) \neq v_1$. Generate the maps $(\mathcal{L}_1, \mathcal{C}, \mathcal{L}_2)$ and $\mathcal{P}$ through the usual key generation algorithm. Generate the sequence $\widetilde{V}_1$ according to the limitations encoded in $\mathbf{v}$, and substitute terms into $\mathcal{C}$, giving $\overline{\mathcal{C}}$. Then,

$$\begin{aligned} \mathsf{sk}^{\eta_4} &= (\widetilde{V}_1, \mathcal{L}_1, \overline{\mathcal{C}}, \mathcal{L}_2, \mathbf{v}), \\ \mathsf{pk}^{\eta_4} &= \mathcal{P}. \end{aligned} \tag{4.4}$$

4.2.2.2 Signature generation.

This step does not change significantly from Rainbow-η. Consider a message M and obtain its digest $\mathbf{h} = \mathcal{H}(M)$. Compute $\mathbf{y} = \mathcal{L}_1^{-1}(\mathbf{h})$, and randomly choose the vinegar variables as encoded in $\mathbf{v}$. Attempt to generate the preimage $\overline{\mathcal{C}}(\mathbf{x}) = \mathbf{y}$ by inverting every layer recursively. If this is not possible, vinegar variables have to be chosen again by the aforementioned procedure. Finally, compute $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{x})$.

4.2.2.3 Signature verification.

This step does not change. If $\mathbf{h} = \mathcal{P}(\mathbf{z})$, then the signature is valid, and otherwise invalid.

We note that there is no need to provide a reconstruction procedure of $\overline{\mathcal{C}}$ as in the case of Rainbow-η_1, which needs a seed, and Rainbow-η_2, that features non-trivial mathematical relations. As seen below, the private key size depends on the Hamming weight of the vinegar threshold.

Proposition 4.2.3. *Consider $\gamma = \mathrm{wt}(\mathbf{v})$. The size of a Rainbow-$\eta_4$ private key, denoted by $sk_\gamma^{\eta_4}$, is*

$$\begin{aligned} \#sk_\gamma^{\eta_4} = &\underbrace{\gamma}_{\widetilde{V}_1} + \underbrace{m^2}_{\mathcal{L}_1} + \underbrace{n^2}_{\mathcal{L}_2} \\ &+ \underbrace{\sum_{k=1}^{u} o_k \cdot \left(\frac{(v_k - \gamma)(v_k - \gamma + 1)}{2} + (v_k - \gamma) \cdot o_k + (v_{k+1} - \gamma) + 1 \right)}_{\overline{\mathcal{C}}} \end{aligned} \tag{4.5}$$

field elements. If the polynomials of $\overline{\mathcal{C}}$ are homogeneous quadratic, the size decreases by $\sum_{k=1}^{u} o_k \cdot (v_{k+1} - \gamma + 1)$ elements.

Proof. We subtract γ terms from every collection of v_k monomials in Equation (2.27), and add $\widetilde{V}_1$ so that the whole preimages can be computed. □

We note that $\#\mathsf{sk}_{v_1}^{\eta_4} = \#\mathsf{sk}^{\eta}$. The representation of $\mathbf{v}$ in the underlying field also needs to be stored. We now expand on the security of our new proposal.

We recall that $\mathbf{v} = (1, \ldots, 1)$ leads to the attack of Subsection 4.1.2. If only a fraction of the vinegar variables are fixed, the remaining polynomial system may be solved in a lower number of operations than the claimed security levels of some parameter sets (SHIM; KOO, 2020).

The authors assume that faults occur when vinegar variables are chosen to sign a message in Rainbow, creating a behaviour analogous to Rainbow-η. However, the intricacy of inducing such faults repeatedly may render the attack impractical (MUS; ISLAM; SUNAR, 2020). According to the model of (SHIM; KOO, 2020), the attacker cannot control the bits flipped through faults. Thus, the absence of information on which vinegar variables are fixed forces the attacker to exhaustively search for such combinations if not enough "weak" signatures are collected.

Therefore, we look at the probabilities associated with obtaining signatures created with repeated vinegar values if only a fraction of those are fixed. The generalized Rainbow-η allows exactly for this choice in the form of the vinegar threshold. We assume that the bits of $\mathbf{v}$ are picked independently, that is, the attacker does not know which and how many vinegar variables are fixed. The distribution which gives this behaviour is presented below.

Definition 4.2.4. Given $n \in \mathbb{N}$, $0 \leq p \leq 1$ and $0 \leq k \leq n$, if the probability mass function of a discrete random variable $\mathbf{X}$ is

$$\Pr[\mathbf{X} = k] = \binom{n}{k} p^k (1-p)^{n-k},$$

then $\mathbf{X}$ follows the *binomial distribution*. The random variable has mean $\mu = np$ and standard deviation $\sigma = \sqrt{np(1-p)}$.

We model the sampling of $\mathbf{v}$ as a binomial distribution with $n = v_1$ and $p = \frac{1}{2}$. In this case, the mean is $\frac{v_1}{2}$ and the standard deviation is $\frac{\sqrt{v_1}}{2}$.

Table 5 shows the approximate probabilities, i.e. $q^{-v_1 + \mathrm{wt}(\mathbf{v})}$, of obtaining signatures with the same $\widetilde{V}_1$ such that $\mathrm{wt}(\mathbf{v}) = \lceil \mu + \beta\sigma \rceil$ for $0 \leq \beta \leq 4$, with parameters from (PETZOLDT, 2013, Tab. 9.8) and (DING et al., 2020, Sec. 4), including a unique instance from the latter in which $q = 16$. We set $\mathrm{q} = q^{-v_1}$ for readability. It is expected that most experiments yield values closer to the center of the PMF, i.e. the mean, and we choose such a quantity to represent a common scenario. Larger Hamming weights, which are desirable for an attacker, are represented by a number of standard deviations away from the mean.

For example, when $q = 256$ and $v_1 = 17$, if half of the bits of $\widetilde{V}_1$ are fixed, there exists a 1 in 5×10^{20} chance that the same vinegar variables are picked for two signatures. It is immediately evident that for lower values of either q or v_1, an attacker is able to mount the attack of Subsection 4.1.2 by simply collecting more signatures. Since there are few possibilities for $\widetilde{V}_1$, it is expected that there is a large number of signatures with the same sequence of vinegar variables.

Moreover, if several vinegar variables are fixed, the complexity of collecting signatures for a successful attack is generally lower than claimed security levels of a parameter set. For

Table 5 – Probabilities of repeating vinegar variables according to the Hamming weight of the vinegar threshold.

q	v_1	$\mathrm{q} \cdot q^{\lceil \mu+0\sigma \rceil}$	$\mathrm{q} \cdot q^{\lceil \mu+1\sigma \rceil}$	$\mathrm{q} \cdot q^{\lceil \mu+2\sigma \rceil}$	$\mathrm{q} \cdot q^{\lceil \mu+3\sigma \rceil}$	$\mathrm{q} \cdot q^{\lceil \mu+4\sigma \rceil}$
256	17	5×10^{-20}	4×10^{-15}	2×10^{-10}	2×10^{-5}	1
256	26	5×10^{-32}	8×10^{-25}	1×10^{-17}	9×10^{-13}	2×10^{-5}
256	36	4×10^{-44}	8×10^{-37}	1×10^{-29}	2×10^{-22}	4×10^{-15}
16	36	2×10^{-22}	9×10^{-19}	4×10^{-15}	1×10^{-11}	6×10^{-8}
256	68	1×10^{-82}	1×10^{-70}	6×10^{-61}	3×10^{-51}	1×10^{-41}
256	96	3×10^{-116}	3×10^{-104}	3×10^{-92}	3×10^{-80}	4×10^{-68}

Source – the author.

example, when $q = 256$ and $v_1 = 68$, the resulting Rainbow scheme has a security level of roughly 2^{192} classical bits (DING et al., 2019, p. 36). An attacker can theoretically forge an equivalent key if $\mathbf{v}$ is less than three standard deviations to the right of the mean. Furthermore, given a sequence of private keys with randomly chosen $\mathbf{v}$, we are able to quantify keys that produce "weak" signatures with high probability. For $q = 256$ and $v_1 = 17$, we have that

$$\begin{aligned} \Pr[\mathbf{X} \geq \mu + 4\sigma] &= \Pr[\mathbf{X} \geq \tfrac{v_1}{2} + 2\sqrt{v_1}] \\ &\approx \Pr[\mathbf{X} = 17] \\ &\approx 7.6294 \times 10^{-6}, \end{aligned}$$

or roughly three in four hundred thousand private keys are reliable targets for attacking, due to a $\mathbf{v}$ with high Hamming weight.

If $u = 1$, i.e. the scheme is a UOV instance, fixed vinegar variables have a slightly reduced effect on the complexity of related attacks. This is due to the nature of the scheme, which needs an increased number of equations and variables to achieve the same level of security when compared to Rainbow. However, the UOV scheme is even less practical than Rainbow due to its larger key and signature sizes. Instead of fixing a small number of vinegar variables to slightly reduce the private key, a user can opt for a different parameter set, or simply use Rainbow without such modifications.

With this analysis, we argue that there are no practical benefits in fixing vinegar variables deliberately. A polynomial-time procedure exists to recover equivalent private keys if Rainbow-η has a single layer and $\mathbf{v} = (1, \ldots, 1)$. Moreover, any vinegar threshold with a non-zero Hamming weight significantly reduces the effort of attackers to break Rainbow instances with commonly used parameters. Therefore, manipulating the randomness of Rainbow signatures in order to reduce private keys is yet another avenue that should not be pursued.

5 CONCLUSION

In this thesis, we analyzed the consequences of limiting the number of possible signatures obtained from a single message. Initially, we sought to combat the premature structural changes to Rainbow private keys to decrease their sizes. To the best of our knowledge, this strategy frequently leads to insecure Rainbow variants. A notable exception is the theory of equivalent keys, which arose from efforts on cryptanalysis of early multivariate public-key cryptosystems, and can be tangentially used to reduce private keys.

To this effect, we introduced a variant of the Rainbow signature scheme that does not limit the private key space. However, private keys are still shortened by substituting values into linear systems early in the key generation algorithm. Consequently, signatures are produced with a lesser degree of randomness, at the expense of a rare key reconstruction event if the resulting linear systems are not solvable. We proposed several methods with varying degrees of performance to solve this specific issue.

We argued that signatures produced with our scheme are statistically indistinguishable from conventional Rainbow signatures, and private keys need reconstruction with negligible probability. Due to our strategy, our method does not impose restrictions on parameters and can be applied to other Rainbow variants to achieve both private and public key size improvements. We obtained reductions of up to 87% in the private key size, and up to 77% in the key pair size when our method is combined with a state-of-the-art strategy that reduces public keys.

The security analysis performed to assert the safety of using our proposal was carried out considering that there were no changes in parameters, thus maintaining the high complexity of known attacks. However, we identified a flaw in our proposal that allows an attacker to efficiently recover an equivalent key by collecting a small number of signatures. We showed how to perform a key recovery attack akin to the method of Kipnis–Shamir to find equivalent keys.

We analyzed the effort needed to collect signatures from an active target that randomly produces signatures with a conventional private key, or the same key reduced with the aid of our proposal. We found out that this strategy thwarts the attacker, but not sufficiently to prevent its eventual progress. Further, we argued that any amount of fixed vinegar variables leads to a theoretically insecure scheme. To this end, we generalized our proposal to introduce a control parameter that rules the level of randomness in the signature generation algorithm, and observed the probabilities associated with attacking signatures with partially fixed vinegar variables.

Despite its simple construction that allows for several modifications, the Rainbow signature scheme is also very prone to introducing subtle weaknesses. Our efforts showed that deviating from customary structural changes to the private key still leads to an insecure scheme. Thus, we restate the importance of rigorously scrutinizing a cryptographic system. Nonetheless, we attempted to provide a comprehensive explanation of our work, hoping that it is beneficial for the community. Further, we have identified several topics to extend our research.

5.1 FUTURE WORK

We observe that there are very few signature schemes based on Rainbow attempting to reduce private and public keys simultaneously. The method of Petzoldt, which is based on generating a structured part of the public key and obtaining the central map later, is the basis of all public-key reduction methods in the literature. Proposals that can be combined with this strategy can reduce the total key pair size, helping storage-constrained devices operate Rainbow in its entirety.

We also argue that there is room for a more robust characterization of the security problems related to Rainbow. The difficulty of the extended isomorphism of polynomials problem (EIP) is not precisely understood. Moreover, the theory of good keys can provide further private-key reductions. However, since it arises from slightly different security problems than EIP, there is limited application of this research to Rainbow.

In the context of our scheme, there are still useful propositions to be answered, even if the security of our proposal is unsatisfactory. Subjecting the scheme to a side-channel attack may help in understanding how a constant-time signature generation algorithm can be achieved, given that the key reconstruction event is discernible.

We assume that the argument bounding the number of signatures needed to attack our proposal can be refined. Our idea is based on choosing a specific tree to easily model connections between linearly independent vector subtractions. However, we expect that the argument follows for all types of trees.

Further work is needed on the complexity estimation of attacks against Rainbow signature schemes with fixed vinegar variables. A pivotal component of such attacks is based on solving random polynomial systems. Algorithms used to this end have unclear asymptotic complexity bounds, which directly influence the security levels of Rainbow instances.

For smaller parameter sets, the chance of accidental reuse of vinegar variables is increased, even if it is only partial. To identify and prevent such collisions, previously used variables can be stored. However, the resulting private key becomes stateful, since it needs to be updated for every signature generated, and its size increases rapidly. Therefore, careful consideration is needed to deal with these shortcomings.

Finally, to the best of our knowledge, the manipulation of vinegar variables in Rainbow throughout means other than fault attacks is nonexistent in the literature. The random generation of vinegar variables by a poor algorithm may be exploited in a similar way to our work. Further, to protect the scheme against such attacks, we suggest the research of methods to prevent tampering of intermediate signing steps, e.g. a checksum, or obtaining vinegar values deterministically from the private key and message.

BIBLIOGRAPHY

ALAGIC, G. et al. **Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process**. [S.l.], 2019.

ALAGIC, G. et al. **Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process**. [S.l.], 2020.

APOSTOL, T. M. **Introduction to Analytic Number Theory**. 1st. ed. [S.l.]: Springer, 2010. ISBN 9781441928054.

BARDET, M. et al. **Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems**. 2020. ArXiv:2002.08322.

BERNSTEIN, D. J.; BUCHMANN, J.; DAHMEN, E. **Post Quantum Cryptography**. 1st. ed. [S.l.]: Springer, 2008. ISBN 9783540887010.

BERNSTEIN, D. J. et al. High-speed high-security signatures. **Journal of Cryptographic Engineering**, v. 2, n. 2, p. 77–89, ago. 2012.

BETTALE, L.; FAUGÈRE, J.-C.; PERRET, L. Solving Polynomial Systems over Finite Fields: Improved Analysis of the Hybrid Approach. In: HOEVEN, J. van der; HOEIJ, M. van (Ed.). **Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation**. [S.l.: s.n.], 2012. p. 67–74.

BEULLENS, W. **New signature schemes based on UOV with smaller public keys**. Dissertação (Mestrado) — Katholieke Universiteit Leuven, jun. 2017.

BEULLENS, W.; PRENEEL, B. Field Lifting for Smaller UOV Public Keys. In: PATRA, A.; SMART, N. P. (Ed.). **Progress in Cryptology – INDOCRYPT 2017**. [S.l.: s.n.], 2017. (Lecture Notes in Computer Science, v. 10698), p. 227–246.

BEULLENS, W.; PRENEEL, B.; SZEPIENIEC, A. Public Key Compression for Constrained Linear Signature Schemes. In: CID, C.; JR., M. J. J. (Ed.). **Selected Areas in Cryptography – SAC 2018**. [S.l.: s.n.], 2018. (Lecture Notes in Computer Science, v. 11349), p. 300–321.

BILLET, O.; GILBERT, H. Cryptanalysis of Rainbow. In: PRISCO, R. de; YUNG, M. (Ed.). **Security and Cryptography for Networks**. [S.l.: s.n.], 2006. (Lecture Notes in Computer Science, v. 4116), p. 336–347.

BITTENCOURT, M. S. P. **Reducing keys in Rainbow-like signature schemes**. Dissertação (Bachelor's thesis) — Universidade Federal de Santa Catarina, nov. 2019.

BORGES, F.; PETZOLDT, A.; PORTUGAL, R. Small private keys for systems of multivariate quadratic equations using symmetric cryptography. **Anais do CNMAC**, v. 4, p. 1085–1091, set. 2012.

BRAEKEN, A.; WOLF, C.; PRENEEL, B. A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes. In: MENEZES, A. (Ed.). **Topics in Cryptology – CT-RSA 2005**. [S.l.: s.n.], 2005. (Lecture Notes in Computer Science, v. 3376), p. 29–43.

CAO, W. et al. Kipnis-Shamir Attack on Unbalanced Oil-Vinegar Scheme. In: BAO, F.; WENG, J. (Ed.). **Information Security Practice and Experience**. [S.l.: s.n.], 2011. (Lecture Notes in Computer Science, v. 6672), p. 168–180.

CHEN, J. et al. Online/offline signature based on UOV in wireless sensor networks. **Wireless Networks**, v. 23, p. 1719–1730, mar. 2016.

CHEN, J.-M.; YANG, B.-Y.; PENG, B.-Y. Tame Transformation Signatures and Topsy-Turvy Hashes. In: **Proceedings of the Second International Workshop for Asian Public Key Infrastructures**. [S.l.: s.n.], 2002. p. 93–100.

CHEN, M.-S. et al. From 5-Pass $\mathcal{MQ}$-Based Identification to $\mathcal{MQ}$-Based Signatures. In: CHEON, J. H.; TAKAGI, T. (Ed.). **Advances in Cryptology – ASIACRYPT 2016**. [S.l.: s.n.], 2016. (Lecture Notes in Computer Science, v. 10032), p. 135–165.

CZYPEK, P.; HEYSE, S.; THOMAE, E. Efficient Implementations of MQPKS on Constrained Devices. In: PROUFF, E.; SCHAUMONT, P. (Ed.). **Cryptographic Hardware and Embedded Systems – CHES 2012**. [S.l.: s.n.], 2012. (Lecture Notes in Computer Science, v. 7428), p. 374–389.

DIFFIE, W.; HELLMAN, M. E. New Directions in Cryptography. **IEEE Transactions on Information Theory**, v. 22, n. 6, p. 644–654, nov. 1976.

DING, J. et al. **Modified Parameters of Rainbow in Response to a Refined Analysis of the Rainbow Band Separation Attack by the NIST Team and the Recent New MinRank attacks**. 2020.

DING, J. et al. **Rainbow - Algorithm Specification and Documentation**. 2017. Round 1 Submission, NIST Post-Quantum Cryptography Standardization Process.

DING, J. et al. **Rainbow - Algorithm Specification and Documentation**. 2019. Round 2 Submission, NIST Post-Quantum Cryptography Standardization Process.

DING, J. et al. Cryptanalysis of The Lifted Unbalanced Oil Vinegar Signature Scheme. In: MICCIANCIO, D.; RISTENPART, T. (Ed.). **Advances in Cryptology – CRYPTO 2020**. [S.l.: s.n.], 2020. (Lecture Notes in Computer Science, v. 12172), p. 279–298.

DING, J.; GOWER, J.; SCHMIDT, D. **Multivariate Public Key Cryptosystems**. 1st. ed. [S.l.]: Springer, 2006. ISBN 9780387322292.

DING, J.; SCHMIDT, D. Rainbow, a New Multivariable Polynomial Signature Scheme. In: IOANNIDIS, J.; KEROMYTIS, A.; YUNG, M. (Ed.). **Applied Cryptography and Network Security**. [S.l.: s.n.], 2005. (Lecture Notes in Computer Science, v. 3531), p. 164–175.

DING, J.; SCHMIDT, D.; YIN, Z. Cryptanalysis of the new TTS scheme in CHES 2004. **International Journal of Information Security**, v. 5, n. 4, p. 231–240, abr. 2006.

DING, J. et al. New Differential-Algebraic Attacks and Reparametrization of Rainbow. In: BELLOVIN, S. M. et al. (Ed.). **Applied Cryptography and Network Security**. [S.l.: s.n.], 2008. (Lecture Notes in Computer Science, v. 5037), p. 242–257.

DORNELLES, M.; LARA, P.; HENRIQUES, F. Performance evaluation and comparison of default and small private key rainbow digital signature scheme for IoT devices. In: **WebMedia '19: Proceedings of the 25th Brazillian Symposium on Multimedia and the Web**. [S.l.: s.n.], 2019. p. 93–96.

DUMMIT, D. S.; FOOTE, R. M. **Abstract Algebra**. 3rd. ed. [S.l.]: Wiley, 2003. ISBN 9780471433347.

DUONG, D. H.; LUYEN, L. V.; TRAN, H. T. N. Choosing subfields for LUOV and lifting fields for rainbow. **IET Information Security**, v. 14, n. 2, p. 196–201, mar. 2020.

DUONG, D. H.; PETZOLDT, A.; TAKAGI, T. Reducing the Key Size of the SRP Encryption Scheme. In: LIU, J. K.; STEINFELD, R. (Ed.). **Information Security and Privacy**. [S.l.: s.n.], 2016. (Lecture Notes in Computer Science, v. 9723), p. 427–434.

DUONG, D. H. et al. Revisiting the Cubic UOV Signature Scheme. In: HONG, S.; PARK, J. H. (Ed.). **Information Security and Cryptology – ICISC 2016**. [S.l.: s.n.], 2016. (Lecture Notes in Computer Science, v. 10157), p. 223–238.

FAUGÈRE, J.-C.; LEVY-DIT-VEHEL, F.; PERRET, L. Cryptanalysis of MinRank. In: WAGNER, D. (Ed.). **Advances in Cryptology – CRYPTO 2008**. [S.l.: s.n.], 2008. (Lecture Notes in Computer Science, v. 5157), p. 280–296.

FURUE, H. et al. A Structural Attack on Block-Anti-Circulant UOV at SAC 2019. In: DING, J.; TILLICH, J.-P. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2020. (Lecture Notes in Computer Science, v. 12100), p. 323–339.

GATHEN, J. von zur. **CryptoSchool**. 1st. ed. [S.l.]: Springer, 2015. ISBN 9783662484234.

GATHEN, J. von zur; GERHARD, J. **Modern Computer Algebra**. 3rd. ed. [S.l.]: Cambrige University Press, 2013. ISBN 9781107039032.

GOLDREICH, O. **Foundations of Cryptography: Volume 2, Basic Applications**. 1st. ed. [S.l.]: Cambridge University Press, 2004. ISBN 9780521830843.

HASHIMOTO, Y. Cryptanalysis of the Quaternion Rainbow. **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences**, E98-A, n. 1, p. 144–152, jan. 2015.

HASHIMOTO, Y. Cryptanalysis of the Multivariate Signature Scheme Proposed in PQCrypto 2013. **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences**, E99.A, n. 1, p. 58–65, jan. 2016.

HASHIMOTO, Y. Weaknesses of cubic UOV and its variants. **Ryukyu Mathematical Journal**, v. 30, p. 1–7, dez. 2017.

HASHIMOTO, Y. Key recovery attack on Circulant UOV/Rainbow. **JSIAM Letters**, v. 11, p. 45–48, mar. 2019.

HASHIMOTO, Y.; IKEMATSU, Y.; TAKAGI, T. Chosen Message Attack on Multivariate Signature ELSA at Asiacrypt 2017. **Journal of Information Processing**, v. 27, p. 517–524, set. 2019.

IPSEN, J.; JERNE, N. K. Graphical Evaluation of the Distribution of Small Experimental Series. **Acta Pathologica Microbiologica Scandinavica**, v. 21, n. 2, p. 343–361, maio 1944.

KIPNIS, A.; PATARIN, J.; GOUBIN, L. Unbalanced Oil and Vinegar Signature Schemes. In: STERN, J. (Ed.). **Advances in Cryptology – EUROCRYPT '99**. [S.l.: s.n.], 1999. (Lecture Notes in Computer Science, v. 1592), p. 206–222.

KIPNIS, A.; SHAMIR, A. Cryptanalysis of the Oil and Vinegar Signature Scheme. In: KRAWCZYK, H. (Ed.). **Advances in Cryptology – CRYPTO '98**. [S.l.: s.n.], 1998. (Lecture Notes in Computer Science, v. 1462), p. 257–266.

KRÄMER, J.; LOIERO, M. Fault Attacks on UOV and Rainbow. In: POLIAN, I.; STÖTTINGER, M. (Ed.). **Constructive Side-Channel Analysis and Secure Design**. [S.l.: s.n.], 2019. (Lecture Notes in Computer Science, v. 11421), p. 193–214.

MULLEN, G. L.; PANARIO, D. **Handbook of Finite Fields**. 1st. ed. [S.l.]: Chapman and Hall/CRC, 2013. ISBN 9781439873786.

MUS, K.; ISLAM, S.; SUNAR, B. **QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme**. 2020.

NAKAMURA, S. et al. **New Complexity Estimation on the Rainbow-Band-Separation Attack**. 2020. Cryptology ePrint Archive, Report 2020/703.

NAKAMURA, S.; WANG, Y.; IKEMATSU, Y. **Analysis on the MinRank Attack using Kipnis-Shamir Method Against Rainbow**. 2020. Cryptology ePrint Archive, Report 2020/908.

NIE, X. et al. Cubic Unbalance Oil and Vinegar Signature Scheme. In: LIN, D.; WANG, X.; YUNG, M. (Ed.). **Information Security and Cryptology**. [S.l.: s.n.], 2015. (Lecture Notes in Computer Science, v. 9589), p. 47–56.

PARK, A. et al. Side-Channel Attacks on Post-Quantum Signature Schemes based on Multivariate Quadratic Equations - Rainbow and UOV -. **IACR Transactions on Cryptographic Hardware and Embedded Systems**, v. 2018, n. 3, p. 500–523, ago. 2018.

PARK, C.-M. Cryptanalysis of Matrix-based UOV. **Finite Fields and Their Applications**, v. 50, p. 209–221, mar. 2018.

PATARIN, J. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: MAURER, U. M. (Ed.). **Advances in Cryptology – EUROCRYPT '96**. [S.l.: s.n.], 1996. (Lecture Notes in Computer Science, v. 1070), p. 33–48.

PATARIN, J. **The Oil and Vinegar Algorithm for Signatures**. 1997. Dagstuhl Seminar 9739.

PATARIN, J.; GOUBIN, L. Trapdoor one-way permutations and multivariate polynomials. In: HAN, Y.; OKAMOTO, T.; QING, S. (Ed.). **Information and Communications Security**. [S.l.: s.n.], 1997. (Lecture Notes in Computer Science, v. 1334), p. 356–368.

PENG, Z.; TANG, S. Circulant Rainbow: A New Rainbow Variant With Shorter Private Key and Faster Signature Generation. **IEEE Access**, v. 5, p. 11877–11886, jun. 2017.

PENG, Z.; TANG, S. Circulant UOV: a new UOV variant with shorter private key and faster signature generation. **KSII Transactions on Internet and Information Systems**, v. 12, n. 3, p. 1376–1395, mar. 2018.

PERLNER, R.; SMITH-TONE, D. **Rainbow Band Separation is Better than we Thought**. 2020. Cryptology ePrint Archive, Report 2020/702.

PETZOLDT, A. **Selecting and Reducing Key Sizes for Multivariate Cryptography**. Tese (Doutorado) — Technische Universität Darmstadt, jul. 2013.

PETZOLDT, A. Efficient Key Generation for Rainbow. In: DING, J.; TILLICH, J.-P. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2020. (Lecture Notes in Computer Science, v. 12100), p. 92–107.

PETZOLDT, A.; BULYGIN, S. Linear Recurring Sequences for the UOV Key Generation Revisited. In: KWON, T.; LEE, M.-K.; KWON, D. (Ed.). **Information Security and Cryptology – ICISC 2012**. [S.l.: s.n.], 2012. (Lecture Notes in Computer Science, v. 7839), p. 441–455.

PETZOLDT, A.; BULYGIN, S.; BUCHMANN, J. A Multivariate Signature Scheme with a Partially Cyclic Public Key. In: FAUGÈRE, J.-C.; CID, C. (Ed.). **International Conference on Symbolic Computation and Cryptography**. [S.l.: s.n.], 2010. p. 229–235.

PETZOLDT, A.; BULYGIN, S.; BUCHMANN, J. CyclicRainbow – A Multivariate Signature Scheme with a Partially Cyclic Public Key. In: GONG, G.; GUPTA, K. C. (Ed.). **Progress in Cryptology – INDOCRYPT 2010**. [S.l.: s.n.], 2010. (Lecture Notes in Computer Science, v. 6498), p. 33–48.

PETZOLDT, A.; BULYGIN, S.; BUCHMANN, J. Selecting Parameters for the Rainbow Signature Scheme. In: SENDRIER, N. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2010. (Lecture Notes in Computer Science, v. 6061), p. 218–240.

PETZOLDT, A.; BULYGIN, S.; BUCHMANN, J. Linear Recurring Sequences for the UOV Key Generation. In: CATALANO, D. et al. (Ed.). **Public Key Cryptography – PKC 2011**. [S.l.: s.n.], 2011. (Lecture Notes in Computer Science, v. 6571), p. 335–350.

PETZOLDT, A. et al. Small Public Keys and Fast Verification for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Public Key Systems. In: PRENEEL, B.; TAKAGI, T. (Ed.). **Cryptographic Hardware and Embedded Systems – CHES 2011**. [S.l.: s.n.], 2011. (Lecture Notes in Computer Science, v. 6917), p. 475–490.

PRESS, W. H. et al. **Numerical Recipes: The Art of Scientific Computing**. 3rd. ed. [S.l.]: Cambridge University Press, 2007. ISBN 9780521880688.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. M. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. **Communications of the ACM**, v. 21, n. 2, p. 120–126, fev. 1978.

SCHNEIER, B. **Memo to the Amateur Cipher Designer**. 1998. Crypto-Gram. Disponível em: https://archive.is/L9DL2.

SEO, H. et al. Small Private Key $\mathcal{MQ}$PKS on an Embedded Microprocessor. **Sensors**, v. 14, n. 3, p. 5441–5458, mar. 2014.

SHIM, K.-A.; KOO, N. Algebraic Fault Analysis of UOV and Rainbow With the Leakage of Random Vinegar Values. **IEEE Transactions on Information Forensics and Security**, v. 15, p. 2429–2439, 2020.

SHIM, K.-A.; KOO, N.; PARK, C.-M. Security Analysis of Improved Cubic UOV Signature Schemes. In: KIM, H.; KIM, D.-C. (Ed.). **Information Security and Cryptology – ICISC 2017**. [S.l.: s.n.], 2017. (Lecture Notes in Computer Science, v. 10779), p. 310–324.

SHIM, K.-A.; PARK, C.-M.; BAEK, Y.-J. Lite-Rainbow: Lightweight Signature Schemes Based on Multivariate Quadratic Equations and Their Secure Implementations. In: BIRYUKOV, A.; GOYAL, V. (Ed.). **Progress in Cryptology – INDOCRYPT 2015**. [S.l.: s.n.], 2015. (Lecture Notes in Computer Science, v. 9462), p. 45–63.

SHIM, K.-A.; PARK, C.-M.; KOO, N. An Existential Unforgeable Signature Scheme Based on Multivariate Quadratic Equations. In: TAKAGI, T.; PEYRIN, T. (Ed.). **Advances in Cryptology – ASIACRYPT 2017**. [S.l.: s.n.], 2017. (Lecture Notes in Computer Science, v. 10624), p. 37–64.

SHIM, K.-A.; PARK, C.-M.; KOO, N. An Efficient MQ-Signature Scheme Based on Sparse Polynomials. **IEEE Access**, v. 8, p. 26257–26265, jan. 2020.

SHOR, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. **SIAM Journal on Computing**, v. 26, n. 5, p. 1484–1509, out. 1997.

SZEPIENIEC, A.; BEULLENS, W.; PRENEEL, B. MQ Signatures for PKI. In: LANGE, T.; TAKAGI, T. (Ed.). **Post Quantum Cryptography**. [S.l.: s.n.], 2017. (Lecture Notes in Computer Science, v. 10346), p. 224–240.

SZEPIENIEC, A.; PRENEEL, B. Block-Anti-Circulant Unbalanced Oil and Vinegar. In: PATERSON, K. G.; STEBILA, D. (Ed.). **Selected Areas in Cryptography – SAC 2019**. [S.l.: s.n.], 2019. (Lecture Notes in Computer Science, v. 11959), p. 574–588.

TAN, Y.; TANG, S. Two Approaches to Build UOV Variants with Shorter Private Key and Faster Signature Generation. In: LIN, D.; WANG, X.; YUNG, M. (Ed.). **Information Security and Cryptology**. [S.l.: s.n.], 2015. (Lecture Notes in Computer Science, v. 9589), p. 57–74.

TAN, Y. et al. Building a new secure variant of Rainbow signature scheme. **IET Information Security**, v. 10, n. 2, p. 53–59, mar. 2016.

TAO, C. **A Method to Reduce the Key Size of UOV Signature Scheme**. 2019. Cryptology ePrint Archive: Report 2019/473.

THOMAE, E. Quo Vadis Quaternion? Cryptanalysis of Rainbow over Non-commutative Rings. In: VISCONTI, I.; PRISCO, R. de (Ed.). **Security and Cryptography for Networks**. [S.l.: s.n.], 2012. (Lecture Notes in Computer Science, v. 7485), p. 361–373.

THOMAE, E. **About the Security of Multivariate Quadratic Public Key Schemes**. Tese (Doutorado) — Ruhr-Universität Bochum, jun. 2013.

THOMAE, E.; WOLF, C. Cryptanalysis of Enhanced TTS, STS and All Its Variants, or: Why Cross-Terms Are Important. In: MITROKOTSA, A.; VAUDENAY, S. (Ed.). **Progress in Cryptology – AFRICACRYPT 2012**. [S.l.: s.n.], 2012. (Lecture Notes in Computer Science, v. 7374), p. 188–202.

TSUJII, S. et al. Proposal of a Signature Scheme Based on STS Trapdoor. In: SENDRIER, N. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2010. (Lecture Notes in Computer Science, v. 6061), p. 201–217.

WOLF, C. **$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Polynomials in Public Key Cryptography**. Tese (Doutorado) — Katholieke Universiteit Leuven, nov. 2005.

WOLF, C.; BRAEKEN, A.; PRENEEL, B. On the security of stepwise triangular systems. **Design, Codes and Cryptography**, v. 40, p. 285–302, mar. 2006.

WOLF, C.; PRENEEL, B. Equivalent keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. **Journal of Mathematical Cryptology**, v. 4, n. 4, p. 375–415, abr. 2011.

YASUDA, T. et al. A Variant of Rainbow with Shorter Secret Key and Faster Signature Generation. In: CHEN, K. et al. (Ed.). **ACM Workshop on Asia Public-Key Cryptography**. [S.l.: s.n.], 2013. p. 57–62.

YASUDA, T.; SAKURAI, K.; TAKAGI, T. Reducing the Key Size of Rainbow Using Non-commutative Rings. In: DUNKELMAN, O. (Ed.). **Topics in Cryptology – CT-RSA 2012**. [S.l.: s.n.], 2012. (Lecture Notes in Computer Science, v. 7178), p. 68–83.

YASUDA, T.; TAKAGI, T.; SAKURAI, K. Multivariate Signature Scheme Using Quadratic Forms. In: GABORIT, P. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2013. (Lecture Notes in Computer Science, v. 7932), p. 243–258.

YASUDA, T.; TAKAGI, T.; SAKURAI, K. Efficient variant of Rainbow using sparse secret keys. **Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications**, v. 5, n. 3, p. 3–13, set. 2014.

YASUDA, T.; TAKAGI, T.; SAKURAI, K. Efficient Variant of Rainbow without Triangular Matrix Representation. In: MAHENDRA, M. S. et al. (Ed.). **Information and Communication Technology**. [S.l.: s.n.], 2014. (Lecture Notes in Computer Science, v. 8407), p. 532–541.

ZAMBONIN, G.; BITTENCOURT, M. S. P.; CUSTÓDIO, R. Handling Vinegar Variables to Shorten Rainbow Private Keys. In: BUCHMANN, J.; NITAJ, A.; RACHIDI, T. (Ed.). **Progress in Cryptology – AFRICACRYPT 2019**. [S.l.: s.n.], 2019. (Lecture Notes in Computer Science, v. 11627), p. 391–408.

ZHANG, M. et al. A Light-Weight Rainbow Signature Scheme for WSN. **Journal of Networks**, v. 7, n. 8, p. 1272–1279, ago. 2012.