Protocolo de Identificação baseado em Polinômios Multivariáveis Quadráticos

Fábio de Salles Monteiro

Dissertação apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre em Ciências

Programa: Ciência da Computação

Orientador: Prof. Dr. Routo Terada

Durante o desenvolvimento deste trabalho o autor
recebeu auxílio financeiro da Marinha do Brasil

São Paulo, novembro de 2012

Protocolo de Identificação baseado em Polinômios Multivariáveis Quadráticos

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 03/12/2012. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Routo Terada (orientador) - IME-USP
- Prof. Dr. Paulo S. L. M. Barreto - EP-USP
- Prof. Dr. Ricardo Dahab - IC/UNICAMP

Agradecimentos

À minha esposa, Paula, agradeço, de maneira especial, pelo amor, dedicação, incentivo e inspiração que me emprestaram forças para concluir com êxito essa jornada.

Aos meus filhos, Guilherme, Beatriz e Bárbara, pelo carinho, as alegrias e mais do que motivação, por encarnarem os melhores motivos do mundo para mim.

Aos meus pais, José Alberto (*in memoriam*) e Sônia por tudo que me ensinaram e mostraram, fazendo me enfrentar os desafios e transpor meus limites.

Ao meu orientador e professor, Routo Terada, por me guiar com seu conhecimento e experiência durante o mestrado.

À Marinha do Brasil pela oportunidade única de realizar este trabalho, ao Centro de Coordenação de Estudos da Marinha em São Paulo e todos os seus integrantes pelo suporte constante que me permitiu uma dedicação total aos afazeres acadêmicos e ao Capitão-de-Corveta Vilc Rufino pela supervisão amistosa e profícua.

Ao professor Paulo S. L. M. Barreto pelas discussões, ideias e conselhos sempre diretos e muito proveitosos.

E a todos os membros do Laboratório de Segurança de Dados do IME/USP pelo apoio e companheirismo.

Resumo

MONTEIRO, F. S. **Protocolo de Identificação baseado em Polinômios Multivariáveis Quadráticos**. 2012. 66 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012.

Os sistemas criptográficos de chave pública amplamente utilizados hoje em dia tem sua segurança baseada na suposição da intratabilidade dos problemas de fatoração de inteiros e do logaritmo discreto, sendo que ambos foram demonstrados inseguros sob o advento dos computadores quânticos. Sistemas criptográficos baseados em Multivariáveis Quadráticas ($\mathcal{MQ}$) utilizam como base o problema $\mathcal{MQ}$, que consiste em resolver um sistema de equações polinomiais multivariáveis quadráticas sobre um corpo finito. O problema $\mathcal{MQ}$ foi provado como sendo NP-completo e até hoje não se conhece algoritmo, nem mesmo quântico, de tempo polinomial que possa resolver o problema, fazendo com que sistemas criptográficos baseados nesta primitiva mereçam ser investigados e desenvolvidos como reais candidatos a proverem nossa criptografia pós-quântica. Durante a CRYPTO'2011 Sakumoto, Shirai e Hiwatari introduziram dois novos protocolos de identificação baseados em polinômios multivariáveis quadráticos, os quais chamamos de $\mathcal{MQID}$-3 e $\mathcal{MQID}$-5, e que em especial e pela primeira vez, tem sua segurança reduzida apenas ao problema $\mathcal{MQ}$. Baseados nestas propostas iremos apresentar uma versão aprimorada do protocolo $\mathcal{MQID}$-3 na qual teremos uma redução da comunicação necessária em aproximadamente 9%.

Palavras-chave: Criptografia Pós-Quântica, Problema $\mathcal{MQ}$, Protocolos de Identificação, Conhecimento-Zero, Chave Pública Multivariável.

Abstract

MONTEIRO, F. S. **Multivariate Quadratic Polynomials Identification Protocol**. 2012. 66 f. Dissertação(Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012.

The public-key cryptography widely used nowadays have their security based on the assumption of the intractability of the problems of integer factorization and discrete logarithm, both of which were proven unsafe in the advent of quantum computers. Cryptographic systems based on Multivariate Quadratic polynomials ($\mathcal{MQ}$) are based on the $\mathcal{MQ}$ problem, which consists in solve a system of multivariate quadratic polynomials over a finite field. The $\mathcal{MQ}$ problem has been proven NP-complete and so far no polynomial time algorithm is known, not even quantum, which would resolve this problem, making worthwhile to be investigated and developed as a real candidate to provide post-quantum cryptography. In CRYPTO'2011 Sakumoto, Shirai and Hiwatari introduced two new identification protocols based on multivariate quadratic polynomials, which we call $\mathcal{MQID}$-3 and $\mathcal{MQID}$-5, in particular, for the first time, their security is based only on the $\mathcal{MQ}$ problem. Using these proposals, we will present an improved version of the protocol $\mathcal{MQID}$-3 that reduces communication by approximately 9%.

Keywords: Post-Quantum Cryptography, $\mathcal{MQ}$ Problem, Identification Protocols, Zero-Knowledge, Multivariate Public-Key.

Sumário

Lista de Abreviaturas

CPQ	Criptografia Pós-Quântica
CZ	Conhecimento-Zero
MPKC	Criptossistema de Chave Pública Multivariável (*Multivariate Public-Key Cryptosystem*)
UOV	Unbalanced Oil and Vinegar
MIA	Matsumoto Imai Scheme A
HFE	Hidden Field Equation
STS	Stepwise Triangular System
ℓ-IC	ℓ-Invertible Cycles

Lista de Símbolos

$\mathbb{F}$ ou $\mathbb{F}_q$	Um Corpo Finito de Galois de ordem q
q	Ordem do Corpo Finito $\mathbb{F}$, ou seja, nº de elementos de $\mathbb{F}$ ($q := \|\mathbb{F}\|$)
$\mathbb{F}^n$	Espaço vetorial de dimensão n sobre $\mathbb{F}$
n	Número de variáveis – $n \in \mathbb{N}$
m	Número de equações – $m \in \mathbb{N}$
d	Grau do sistema de equações – $d \in \mathbb{N}$ (quando não dito contrário usaremos $d = 2$)
$p(x_1, ..., x_n)$	Um polinômio de grau d em n variáveis sobre $\mathbb{F}$
$\mathcal{P} = (p_1, \ldots, p_m)$	Um sistema de m polinômios de grau d em n variáveis sobre $\mathbb{F}$
$a \in_R \mathcal{D}$	Um valor a pertencente ao domínio $\mathcal{D}$ é escolhido aleatoriamente, com igual probabilidade, entre todos os elementos de $\mathcal{D}$
$\{0,1\}^*$	Conjunto de todas as strings binárias de qualquer tamanho arbitrário finito
$\{0,1\}^n$	Conjunto de todas as strings binárias de tamanho n
$A \times B$	Conjunto de todos os pares (a, b), onde $a \in A$ e $b \in B$
$F : A \mapsto B$	uma função F tendo como domínio o conjunto A e imagem o conjunto B
$a\|\|b$	Concatenação binária de a e b
lg	O mesmo que $\log_2$

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A palavra criptografia deriva do grego kryptós – escondido ou oculto, e gráphein – escrita, dando o significado original do termo, que referia-se exclusivamente a comunicações secretas, normalmente utilizadas em meios militares. Porém, a criptografia moderna que estudamos e utilizamos hoje em dia vai muito além do seu propósito inicial, lidando, além do sigilo, também com a integridade e a legitimidade de informações, que podem ser textos, documentos, programas ou qualquer outra forma de dados eletrônicos. A criptografia moderna deixou há muito de ser uma ferramenta de uso exclusivamente militar para se tornar parte do dia a dia de todas as pessoas. Ela está nos cartões de banco, nos celulares, nas assinaturas digitais de documentos eletrônicos, na conexão a sítios seguros na internet e nas urnas eletrônicas de nossas eleições garantindo nossa democracia, apenas para citar alguns lugares onde as pessoas utilizam a criptografia, na maioria das vezes sem mesmo saber. Vemos assim que a criptografia moderna se tornou uma tecnologia imprescindível para a humanidade.

1.1 Caracterização do Problema

Como vimos, temos atualmente uma grande dependência da criptografia e um problema que se apresenta é que os sistemas criptográficos de chave pública amplamente utilizados hoje em dia tem sua segurança baseada na suposição da intratabilidade dos problemas de fatoração de inteiros, no caso de sistemas RSA, e do logaritmo discreto, em sistemas ElGamal ou de Curvas Elípticas, sendo possível resolver tais problemas em tempo polinomial com algoritmos quânticos [58], o que tornaria inseguros nossos sistemas criptográficos atuais quando possuirmos computadores quânticos com a capacidade adequada. É verdade que não há no curto prazo nenhuma certeza sobre o desenvolvimento e a adoção da computação quântica em larga escala. Existe uma corrente que antecipa o colapso da criptografia moderna como uma catástrofe iminente que precisamos combater, enquanto outros dizem que isso nunca se tornará realidade e portanto não teríamos que despender esforços combatendo "moinhos". Preferimos crer que qualquer opinião extre-

mada tende a ignorar a necessidade de lidarmos com múltiplos cenários, evitando apostas contundentes, que como o próprio nome diz, são apenas apostas. Na verdade a evolução dos computadores quânticos não deveria ser vista como a fronteira de obsolescência dos atuais esquemas baseados em criptografia assimétrica, ou de chave pública. De fato, já existe a premente necessidade de buscarmos a atualização, entre outros, dos esquemas de assinatura digitais, para que não tenhamos no futuro que conviver com uma eventual situação de insegurança jurídica em virtude de documentos assinados décadas antes, mas que devido a evolução do poder computacional não mais poderiam ter sua autenticidade e integridade garantidos, acarretando o comprometimento de documentos eletrônicos assinados hoje em dia, por exemplo.

Dentro do universo da criptografia moderna, os protocolos de identificação são peças fundamentais para a construção de mecanismos de autenticação e assinaturas digitais. Quando baseados no modelo de chave pública, sua segurança depende da hipótese de intratabilidade de um determinado problema computacional. O problema de encontrar solução para um sistema de equações polinomiais multivariáveis quadráticas sobre um corpo finito, conhecido por Problema $\mathcal{MQ}$, reúne propriedades de interesse para o desenvolvimento de primitivas criptográficas, visto que é um problema NP-completo [29], não se conhecendo, até hoje, solução de tempo polinomial, nem mesmo no modelo computacional quântico [44].

1.2 Criptografia Pós-Quântica

Convencionou-se chamar de Criptografia Pós-Quântica a área de pesquisa onde são desenvolvidos sistemas criptográficos baseados em problemas intratáveis em computadores quânticos e onde, também, determina-se a complexidade quântica das hipóteses de intratabilidade. Podemos destacar como as principais classes de criptossistemas aceitas como pós-quânticas [5]: Códigos Corretores de Erros, Hash, Reticulados e Chave Pública Multivariável, sendo importante destacar que não possuímos ainda uma comprovação fática da intratabilidade dos problemas bases desses criptossistemas em um computador quântico, como bem destaca a conclusão de Buchmann *et al.* [9]. Dentre essas quatro classes citadas, iremos explorar neste trabalho os Criptossistemas de Chave Pública Multivariável, que foram originados do trabalho de Matsumoto e Imai [41] e são conhecidos como MPKC (acrônimo da nomenclatura em inglês) [19, 64, 66].

1.3 Propostas originais de Sakumoto, Shirai e Hiwatari

Durante a conferência CRYPTO 2011, Sakumoto, Shirai e Hiwatari apresentaram uma nova forma de construir parâmetros públicos e secretos com base no problema $\mathcal{MQ}$, de modo a eliminar a hipótese de dificuldade do problema de isomorfismo de polinômios (IP), presente em vários outros trabalhos [55]. Os autores apresentaram dois protocolos de identificação de conhecimento-zero baseados exclusivamente no problema $\mathcal{MQ}$, aplicando uma técnica de divisão do segredo e a forma polar (com propriedade de bilinearidade) de uma função $\mathcal{MQ}$. Para reduzir ao mínimo possível a comunicação, eles utilizaram ainda uma técnica criada originalmente por Jacques Stern [60], que consiste em enviar apenas um hash de todos os compromissos no primeiro passo e depois incluir em *Rsp* aqueles compromissos que não podem ser verificados, para que juntamente com os compromissos reconstituídos pelo verificador seja possível efetuar um novo hash para validar o recebido inicialmente. No artigo de Sakumoto, Shirai e Hiwatari todos os dados atinentes ao tamanho da comunicação do protocolo consideram a utilização desta técnica e por isso iremos assumir que ela será obrigatoriamente utilizada, visto que nosso objetivo principal é, de fato, a diminuição do tamanho da comunicação do protocolo, objetivando atender aplicações onde o consumo de energia é crítico e a complexidade de comunicação requer mais energia do que a complexidade de processamento local, como em redes de sensores sem fio, por exemplo. Chamaremos o protocolo em 3 passos de Sakumoto, Shirai e Hiwatari de $\mathcal{MQID}$-3 e, consequentemente, de $\mathcal{MQID}$-5 o protocolo original em 5 passos.

1.4 Objetivos

Os objetivos principais deste trabalho são:

- análise da proposta de Sakumoto, Shirai e Hiwatari de um novo modelo MPKC, idealizado para ter sua segurança reduzida ao problema $\mathcal{MQ}$;
- a apresentação de um novo protocolo de identificação $\mathcal{MQID}$-3 Aprimorado, com foco na redução da comunicação necessária; e
- o exame do modelo consagrado anteriormente de criptossistemas de chave pública multivariável.

1.5 Contribuições

A principal contribuição deste trabalho é a construção de uma versão aprimorada do protocolo de identificação $\mathcal{MQID}$-3, apresentado durante a CRYPTO'2011 por Sakumoto, Shirai e Hiwatari [55], utilizando uma construção que diminui a probabilidade de personificação em uma rodada de 2/3 para 1/2, elevando o nível de segurança do protocolo e diminuindo o tamanho da comunicação necessária em uma execução. Com os parâmetros sugeridos originalmente nossa proposta apresenta uma diminuição de 9,5% no tamanho da comunicação, considerando do mesmo modo que Sakumoto *et al.* a utilização da técnica sugerida por Stern [60] para reduzir o número de compromissos trafegados, enviando apenas um hash de todos os compromissos no primeiro passo do protocolo e depois incluindo na resposta do provador apenas aqueles compromissos que não podem ser reconstituídos, para que juntamente com os compromissos validados pelo verificador seja possível efetuar um novo hash para comparar com o recebido inicialmente. Nosso protocolo preserva o tamanho dos parâmetros do sistema e também independe da hipótese de dificuldade do problema de isomorfismo de polinômios.

1.6 Organização do Trabalho

No Capítulo 2, apresentamos os conceitos e notações que precisaremos para o desenvolvimento do trabalho. No Capítulo 3 detalhamos o Problema $\mathcal{MQ}$ e seu uso em criptografia de chave pública. Apresentamos as funções $\mathcal{MQ}$-Trapdoor básicas, revisamos o atual modelo MPKC, bem como a formulação de Sakumoto, Shirai e Hiwatari para um novo modelo MPKC baseado apenas no problema $\mathcal{MQ}$. Encerramos este Capítulo com o estado-da-arte da criptanálise MPKC. Finalmente, no Capítulo 4 vamos apresentar nossa proposta de um protocolo $\mathcal{MQID}$-3 Aprimorado com as necessárias análises e comparações destacando as melhorias conseguidas com a nossa versão. No Capítulo 5 discutimos algumas conclusões obtidas neste trabalho e propomos trabalhos que podem ser derivados deste.

Capítulo 2

Conceitos Preliminares

Para o desenvolvimento deste trabalho utilizamos muitas ferramentas advindas de conceitos básicos da matemática e da criptologia, os quais passamos a explorar neste capítulo. Embora muitos leitores possam sentir-se confortáveis em pular este capítulo, consideramos importante destacar e explicar os conceitos aqui contidos para facilitar o entendimento do trabalho principalmente pelos discentes.

2.1 Fundamentos Matemáticos

Nesta seção estabeleceremos conceitos matemáticos importantes que irão nos ajudar na compreensão dos capítulos seguintes. Não intencionamos, nem poderíamos, esgotar toda a teoria matemática interligada aos assuntos discutidos, para referências mais completas e detalhadas sugerimos [59], [33] e [35].

2.1.1 Grupo

Um Grupo $\mathbb{G}$ é composto pelo par (G, $\bullet$), onde G é um conjunto não vazio e $\bullet$ é uma operação binária definida sobre os elementos de G, sendo que $\mathbb{G}$ possui as seguintes propriedades:

1. **Associatividade**: A operação $\bullet$ é associativa, que quer dizer que, a ordem pela qual agrupamos as operações $\bullet$, quando ela aparece mais de uma vez em uma expressão, é irrelevante. Ou seja, $\forall x, y, z \in \mathrm{G} : (x \bullet y) \bullet z = x \bullet (y \bullet z)$

2. **Elemento Neutro**: Existe um elemento $i \in \mathrm{G}$, chamado de *identidade*, tal que quando ele for um dos dois elementos envolvidos na operação, o resultado será igual ao outro elemento, isto é, $\forall x \in \mathrm{G} : x \bullet i = i \bullet x = x$.

3. **Elemento Simétrico ou Inverso**: Para cada elemento $x \in \mathrm{G}$ existe um elemento $\overline{x} \in \mathrm{G}$, chamado de *inverso* de x, tal que $x \bullet \overline{x} = \overline{x} \bullet x = i$, sendo i o elemento neutro supracitado.

Grupo Abeliano - Quando a operação $\bullet$ é comutativa, ou seja, $\forall x, y \in \mathrm{G} : x \bullet y = y \bullet x$, dizemos que o grupo é comutativo ou abeliano.

Na tabela 2.1 vemos os símbolos que estamos acostumados de acordo com os tipos de grupos mais utilizados.

Grupo	Operação $\bullet$	Elemento neutro i	Inverso de x
Multiplicativo	$*$	1	x^{-1}
Aditivo	$+$	0	$-x$

Tabela 2.1: *Grupos*

Chamamos de ordem de um grupo a quantidade de elementos componentes do conjunto G, ordem = |G|. Se G é infinito então o grupo possui ordem infinita. Normalmente na literatura, os autores se referem simplesmente a G para citar tanto o grupo propriamente dito, quanto o conjunto que o compõe.

Como consequência das propriedades citadas na definição temos que:

1. G possui apenas um elemento identidade, senão vejamos, caso houvessem dois elementos $i, i' \in \mathrm{G}$, sendo ambos identidade, então pela propriedade 2 da nossa definição de Grupo teríamos $i = i \bullet i' = i'$, o que não é possível.

2. Cada elemento de G possui um e apenas um inverso, como podemos verificar examinando a seguinte igualdade, onde $\overline{x}$ e $\overline{x}'$ seriam dois elementos simétricos de x, teríamos pela definição que $\overline{x} = \overline{x} \bullet i = \overline{x} \bullet (x \bullet \overline{x}') = (\overline{x} \bullet x) \bullet \overline{x}' = i \bullet \overline{x}' = \overline{x}'$, ou seja $\overline{x} = \overline{x}'$, tendo x apenas um inverso, c.q.d.

2.1.2 Anel

Um Anel $\mathbb{R}$ é composto pela tripla (R, *, +), onde R é um conjunto não vazio e + e * são duas operações binárias definidas sobre os elementos de R, sendo que $\mathbb{R}$ possui as seguintes propriedades:

1. **Grupo Abeliano Aditivo**: O par (R, +) forma um grupo abeliano conforme descrito na seção anterior.

2. **Associatividade de ***: A operação * é associativa. A ordem pela qual agrupamos as operações de multiplicação, quando ela aparece mais de uma vez em uma expressão, é irrelevante, ou seja, $\forall x, y, z \in \mathrm{R} : (x * y) * z = x * (y * z)$

3. **Distributividade**: A operação * é distributiva sobre +, isto é, $\forall x, y, z \in \mathrm{R} : x * (y + z) = (x * y) + (x * z)$

Anel Comutativo - Quando a operação * também é comutativa, ou seja, $\forall x, y \in \mathrm{R} : x * y = y * x$, dizemos que o anel é comutativo.

Anel com Identidade - Chamamos de Anel com Identidade quando existe um elemento neutro da multiplicação, $1 \in \mathrm{R}$, tal que quando ele for um dos dois elementos envolvidos na operação *, o resultado será igual ao outro elemento, isto é, $\forall x \in \mathrm{R} : x * 1 = 1 * x = x$.

2.1.3 Corpo

Utilizando nossas definições anteriores, dizemos que um corpo $\mathbb{F}$ é um anel comutativo com identidade em que todo elemento diferente de 0 possui um elemento inverso com relação à multiplicação, ou seja, $\forall x \in \mathrm{F}\backslash\{0\}\ \exists x^{-1} \in \mathrm{F} : x * x^{-1} = 1$.

Chamamos então de Corpo finito quando o conjunto dos elementos é finito. Corpos finitos também são chamados corpos de Galois em honra ao matemático francês Évariste Galois. Em criptografia normalmente utilizamos Corpos finitos, os quais somente podem possuir ordem no formato p^n, sendo p um número primo, que chamamos de característica do Corpo, e n um inteiro positivo. Podemos ver a demonstração dessa condição em [59, Teoremas 7.5 e 7.7]. Quando o Corpo é de ordem infinita dizemos que possui característica zero.

2.1.4 Espaço Vetorial

Um Espaço Vetorial $\mathbb{V}$ é composto pela tupla (V, +, *, $\mathbb{F}$), onde V é um conjunto não vazio, + e * são duas operações binárias, sendo + definida sobre os elementos de V e * definida por $\mathbb{F} \times \mathrm{V} \mapsto \mathrm{V}$ e $\mathbb{F}$ é um corpo. Chamamos os elementos de V de vetores e os elementos do conjunto de $\mathbb{F}$ de escalares. Devemos ainda observar que o espaço vetorial $\mathbb{V}$ tem que possuir as seguintes propriedades:

1. **Grupo Abeliano Aditivo**: O par (V, +) forma um grupo abeliano conforme descrito anteriormente.

2. **Associatividade de ***: A operação * é associativa. A ordem pela qual agrupamos as operações de multiplicação, quando ela aparece mais de uma vez em uma expressão, é irrelevante, ou seja, $\forall x, y \in \mathbb{F}$ e $\forall v \in \mathrm{V} : x * (y * v) = (x * y) * v$.

3. **Distributividade**: A operação * é distributiva sobre +, isto é, $\forall x \in \mathbb{F}$ e $\forall u, v \in \mathrm{V} : x * (u + v) = (x * u) + (x * v)$.

4. **Distributividade sobre adição no Corpo** $\mathbb{F}$: A operação * é distributiva sobre adição no Corpo $\mathbb{F}$, isto é, $\forall x, y \in \mathbb{F}$ e $\forall v \in \mathrm{V} : (x + y) * v = (x * v) + (y * v)$.

5. **Elemento Neutro de ***: Seja 1 o elemento neutro ou identidade de $\mathbb{F}$ então quando ele for um dos dois elementos envolvidos na operação *, o resultado será igual ao elemento vetor, isto é, $\forall v \in \mathrm{V} : 1 * v = v$.

Um exemplo básico e recorrente de espaço vetorial seria o conjunto $\mathbb{R}^2$ sobre o corpo dos números reais, com as operações usuais de adição e multiplicação. É comum designarmos um espaço vetorial pelo símbolo do corpo elevado ao número de elementos de cada vetor, *e.g.*, $\mathbb{F}^n$.

2.1.5 Bilinearidade

Seja $\mathbb{V}$ um espaço vetorial (V, +, *, $\mathbb{F}$), uma função B: V $\times$ V $\mapsto$ $\mathbb{F}$ é dita bilinear quando cumpre os seguintes axiomas $\forall x, y, z \in \mathrm{V}$:

1. $\mathrm{B}(x + y, z) = \mathrm{B}(x, z) + \mathrm{B}(y, z)$
2. $\mathrm{B}(x, y + z) = \mathrm{B}(x, y) + \mathrm{B}(x, z)$
3. $\mathrm{B}(y * x, z) = \mathrm{B}(x, y * z) + y * \mathrm{B}(x, z)$

2.2 Principais notações utilizadas

Na nossa Lista de Símbolos, constante da página xiii, temos a relação completa de notações e símbolos que utilizamos, porém, com a finalidade de facilitar a leitura, destacamos aqui as principais notações que serão mais frequentemente vistas neste trabalho. Durante esta dissertação utilizaremos $\mathbb{F}$ ou $\mathbb{F}_q$ para designar um corpo de Galois de ordem q. Quando utilizamos $\mathbb{F}^n$ estamos falando do espaço vetorial de dimensão n sobre $\mathbb{F}$. Em nossa notação, $a \in_R \mathcal{D}$ determina que um valor a pertencente ao domínio $\mathcal{D}$ é escolhido aleatoriamente, com igual probabilidade, entre todos os elementos pertencentes a $\mathcal{D}$.

2.3 Probabilidade Desprezável

Para formalizarmos várias definições precisamos antes do conceito de probabilidade desprezável. É intuitivo imaginar que algo com probabilidade desprezável tem uma chance muito remota de ocorrer, tão remota que podemos ignorar. Porém, apenas essa ideia, sem uma definição específica, provoca inevitavelmente uma dubiedade no entendimento. Damgard e Nielsen [16] citam que é possível estabelecer de uma forma concreta uma probabilidade desprezável como sendo qualquer acontecimento que ocorra com uma probabilidade menor que um valor determinado, como 2^{-30}. Porém, eles, assim como nós, pensam que é preferível utilizarmos uma definição assintótica.

Probabilidade Desprezável $\epsilon - \epsilon(l)$ é desprezável em l se para qualquer polinômio p, $\epsilon(l) \leq 1/p(l)$ para todo l grande o suficiente.

2.4 Funções Unidirecionais

Funções unidirecionais (*one-way functions*) desempenham um importante papel na criptografia moderna. Elas possuem a característica de não serem inversíveis, ou melhor dizendo, serem difícil de inverter, conforme iremos melhor definir abaixo. Sua existência é aceita pela maioria, apesar de não ter sido possível comprova-la até o presente momento. Para formalizarmos nossa definição de função unidirecional, e posteriormente também a de função unidirecional com segredo, utilizamos como base a definição de Patarin e Goubin [47].

Seja $f : \mathcal{D} \mapsto \mathcal{I}$ uma função. f é dita unidirecional se possuir as duas propriedades seguintes:

Fácil de calcular – Dado $x \in \mathcal{D}$, é possível calcular em tempo polinomial $y = f(x)$.

Difícil de inverter – Dado $y \in_R \mathcal{I}$, não se conhece algoritmo de tempo polinomial em $|y|$ que calcule $x \in \mathcal{D}$ tal que $f(x) = y$.

2.4.1 Função de Hash

Quando uma função unidirecional possui um tamanho de saída fixo, independente do tamanho da entrada, isto é, $h : \{0,1\}^* \mapsto \{0,1\}^l$, costumamos chama-la de função de hash. A definição que utilizamos aqui corresponderia à apresentada por Menezes *et al.* [42] como sendo uma função de hash unidirecional, havendo antes, no livro citado, uma definição mais genérica de função de hash, sem o requisito das propriedades de função unidirecional. Para nós, no entanto, interessa apenas a definição que formulamos acima. Funções de hash são muito utilizadas para verificações de integridade e autenticação de mensagens, com sua saída servindo como um valor de conferência para validar se um determinado string de dados, que pode ser um arquivo, um texto ou qualquer outra informação, permanece inalterado. Veremos um pouco mais a frente a utilização de funções de hash para a construção de esquemas de comprometimento e também podemos citar seu destacado papel nos esquemas de assinatura digital, como exemplo de sua utilidade prática.

Chamamos de colisão quando temos $a, b \in \{0,1\}^*, a \neq b$ e $h(a) = h(b)$. Visto que a e b podem possuir tamanhos maiores que l, ou seja, que a saída da função de hash, teremos $|a|, |b| > l$ e podemos ver que colisões obrigatoriamente existem. Uma função de hash é dita resistente à colisão quando não se conhece algoritmo de tempo polinomial que possibilite encontrar essas colisões.

Em [43] pode ser visto um breve exame (*survey*) resumido sobre teoria, ataques e aplicações de funções de hash. Porém, precisamos aqui destacar ainda o ataque do aniversário (*birthday attack*), o que faremos utilizando o que encontramos muito bem sintetizado em [36, capítulo 4.6.3]. Se o tamanho da saída de uma função de hash é de l bits então o ataque do aniversário encontra com alta probabilidade uma colisão usando $O(2^{l/2})$

execuções da função de hash. Daí decorre a conclusão que para alcançar uma segurança de κ bits precisamos de uma função de hash com saída de pelo menos 2κ bits.

2.4.2 Função Unidirecional com segredo (trapdoor)

O conceito de função unidirecional com segredo, também chamada de função trapdoor, foi esboçado pela primeira vez juntamente com a própria apresentação da criptografia assimétrica por Diffie e Hellman [17]. Abaixo apresentamos então uma definição de função unidirecional com segredo, simples e consistente com o nosso contexto.

Seja $f : \mathcal{D} \mapsto \mathcal{I}$ uma função. f é dita uma função unidirecional com segredo se:

1. f é uma função unidirecional.
2. Existe uma informação secreta t tal que, dado $y \in_R \mathcal{I}$ e t, é possível calcular em tempo polinomial $x \in \mathcal{D}$ tal que $f(x) = y$.

2.5 Introdução a Esquemas de Comprometimento

Um esquema de comprometimento (*Commitment Scheme*) tem por objetivo esconder temporariamente um valor que não pode ser alterado.

Podemos ilustrar o funcionamento de um esquema de comprometimento com um remetente que tranca uma mensagem em uma caixa e a entrega para um destinatário. Podemos ver que a mensagem está escondida, visto que se encontra trancada no interior da caixa, como queríamos desde o princípio. Também percebemos que não é mais possível ao remetente alterar essa mensagem, pois já entregou a caixa ao destinatário. Posteriormente, quando convier ao remetente, ele entrega a chave para o destinatário possibilitando que ele confira a mensagem. Vemos assim que um esquema de comprometimento funciona em duas fases distintas. Na primeira, que chamamos de fase de comprometimento, um remetente irá se "comprometer" com uma mensagem específica, entregando ao destinatário a "caixa trancada". Uma segunda fase, que chamamos de verificação, ocorrerá quando o remetente entregar ao destinatário a "chave" que possibilitará verificar a mensagem original.

Para formalizarmos este conceito, dizemos que um esquema de comprometimento $\mathcal{C}om$ é uma tupla com dois algoritmos chamados Empenha e Abre, onde Empenha recebe uma mensagem $msg \in \mathcal{M}$, sendo $\mathcal{M}$ o espaço de mensagens, e retorna um par c e d, onde c é um valor de compromisso e d é a chave para recuperar msg com a utilização do algoritmo Abre e o próprio c. Abre retorna msg ou então $\perp$, caso c seja inválido. Em um esquema de comprometimento correto deveremos ter que Abre(Empenha(msg)) = msg. Precisamos ainda que este esquema de comprometimento possua as seguintes propriedades.

Ocultação (*hiding*) – Seja $\mathcal{C}om$ um esquema de comprometimento, dizemos que ele possui ocultação se é computacionalmente difícil descobrir qualquer informação sobre msg a partir de c.

Vinculação (*binding*) – Seja $\mathcal{C}om$ um esquema de comprometimento, dizemos que ele possui vinculação se, efetuado Empenha(msg_1) → c, d, é computacionalmente difícil formular um valor d' tal que Abre(c,d') = msg_1 ou Abre(c, d') $\neq \perp$.

Em [34] pode ser visto um modelo de esquema de comprometimento, prático e comprovadamente seguro, que é construído a partir da utilização de funções de hash.

2.6 Introdução a Esquemas de Identificação

Esquemas de identificação são peças fundamentais para a construção de mecanismos de autenticação e assinaturas digitais. Seu objetivo é possibilitar que uma entidade $\mathcal{B}eto$, neste contexto também chamado de verificador, possa validar com segurança a identidade de uma outra entidade $\mathcal{A}lice$, também chamada de provadora. Obviamente, existem diversas maneiras de se alcançar o objetivo proposto. Podemos, por exemplo, considerar o login de um usuário em um sistema operacional como sendo um esquema de identificação baseado em chave secreta, onde o usuário é o provador e o SO é o verificador. Neste exemplo, o verificador simplesmente conhece a chave secreta do provador e faz a checagem validando-o. Em nosso trabalho no entanto, vamos focar apenas em esquemas de identificação baseados no modelo de chave pública, mais especificamente vamos utilizar apenas esquemas de identificação de conhecimento-zero, por isso, a partir de agora, quando nos referirmos a esquema de identificação, estamos falando de um esquema de conhecimento-zero.

Para formalizarmos nossa definição dizemos que um esquema de identificação $\mathcal{ID}$ é uma tupla de algoritmos composta por Setup, Gen, P e V. Onde Setup recebe um parâmetro de segurança 1^κ e devolve um parâmetro de sistema *param*. Gen é o algoritmo de geração de chaves que utiliza *param* e retorna um par de chaves sk e pk, onde sk é a chave privada e pk a chave pública. P e V constituem um protocolo interativo de identificação, onde P é executado por $\mathcal{A}lice$ (provadora), utilizando sua chave privada, e V é utilizado por $\mathcal{B}eto$ (verificador), com acesso a chave pública de $\mathcal{A}lice$. A seguir estabelecemos as propriedades requeridas para o nosso esquema de identificação:

Correção (*completeness*) – Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele satisfaz a noção de correção se, dados uma provadora honesta $\mathcal{A}lice$ e um verificador honesto $\mathcal{B}eto$, a execução do protocolo resulta no aceite da identidade de $\mathcal{A}lice$ por $\mathcal{B}eto$, exceto por uma probabilidade desprezável, ou seja:
$\Pr[\mathrm{V}(pk_{\mathcal{A}lice}, Cmt \| Ch \| \mathrm{P}(sk_{\mathcal{A}lice}, Cmt \| Ch)) = 1] = (1 - \epsilon)$.

Vale citar o caso específico quando $\epsilon = 0$ e então dizemos que o protocolo possui correção perfeita ou que é perfeitamente correto, valendo o seguinte:

$$\Pr[\mathrm{V}(pk_{Alice}, Cmt\|Ch\|\mathrm{P}(sk_{Alice}, Cmt\|Ch)) = 1] = 1$$

Solidez (*soundness*) – Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele possui solidez se, nenhum falso provador puder convencer um verificador honesto a aceitar sua interação como sendo de uma outra entidade, exceto com uma probabilidade desprezável. Isto é, se um adversário $\mathcal{C}arlos$ tentar personificar $\mathcal{A}lice$ perante $\mathcal{B}eto$, sua chance de sucesso será desprezível ou, formalmente, menor que ϵ.

$$\Pr[\mathrm{V}(pk_{Alice}, Cmt\|Ch\|\mathrm{P}(sk_{falsa}, Cmt\|Ch)) = 1] < \epsilon$$

Conhecimento-Zero (*zero-knowledge*) – Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele é de conhecimento-zero (CZ) se, caso a afirmação seja verdadeira, nenhum verificador aprende outra coisa senão este fato. Este conceito é normalmente formalizado demonstrando-se que um verificador trapaceiro com auxílio de um simulador, utilizando somente informações públicas, pode produzir uma transcrição do protocolo que "parece"uma interação entre um provador honesto e o verificador.

CZ Perfeito – Quando a distribuição de probabilidades do simulador e de uma interação entre um provador honesto e o verificador são exatamente iguais.

CZ Estatístico – Quando a distribuição de probabilidades do simulador e de uma interação entre um provador honesto e o verificador são desprezáveis.

CZ Computacional – Quando a distribuição de probabilidades do simulador e de uma interação entre um provador honesto e o verificador são indistinguíveis computacionalmente em tempo polinomial.

Quando o esquema de identificação é composto de 3 passos, conforme demonstrado na figura 2.1, ele é chamado de canônico por Abdalla *et al.* [1] que especifica esta condição como um dos requisitos para possibilitar a geração de um esquema de assinatura digital, seguro no modelo do oráculo aleatório [4], utilizando-se o paradigma da transformação Fiat-Shamir, que iremos explicar adiante na seção 2.8.

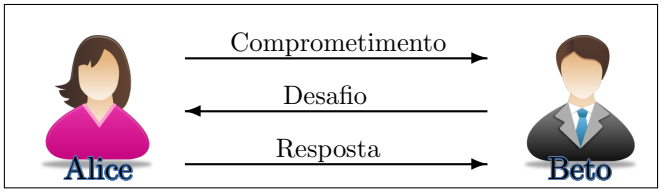

Figura 2.1: *Passos de um Protocolo Canônico*

Abaixo apresentamos uma classificação dos tipos de ataque a Esquemas de Identificação:

Ataque de chave pública (*Key-Only attack*) - O adversário tem acesso apenas a chave pública.

Ataque passivo (*Passive attack*) - O adversário tem acesso a um determinado número de transcrições legítimas de identificações realizadas por $\mathcal{A}lice$ e $\mathcal{B}eto$.

Ataque ativo (*Active attack*) - O adversário age como verificador, interagindo com o provador.

Ataque ativo simultâneo (*Concurrent attack*) - O adversário age como verificador realizando diversas validações de identidade simultâneas do provador para tentar conseguir alguma vantagem.

No Capítulo 4 estudaremos os detalhes necessários do protocolo $\mathcal{MQID}$-3, de Sakumoto, Shirai e Hiwatari [55], antes de apresentar nossa proposta para uma versão aprimorada e com uma comunicação total menor.

2.7 Introdução a Esquemas de Assinatura Digital

Podemos definir um esquema de assinatura digital como uma tupla de algoritmos composta por Setup, Gen, Assina e Verifica. Setup recebe um parâmetro de segurança 1^κ e devolve um parâmetro de sistema *param*. Gen é o algoritmo de geração de chaves que utiliza *param* e retorna um par de chaves sk e pk, onde sk é a chave privada e pk a chave pública de um usuário. Assina é um algoritmo probabilístico que utiliza uma chave privada sk juntamente com uma mensagem msg e devolve uma assinatura σ. Verifica é um algoritmo determinístico que retorna um valor dec $\in \{0, 1\}$ a partir da validação de um σ e uma chave pública pk, sendo 1 para uma assinatura legítima e 0 caso contrário. $Verifica(pk_{\mathcal{A}lice}, Assina(sk_{\mathcal{A}lice}, msg)) = 1$

Pointcheval e Stern [51] definiram como genérico o esquema de assinatura digital no qual σ é formado por uma tripla (σ_1, h, σ_2), sendo h o hash de (msg, σ_1) e σ_2 dependente apenas de σ_1, msg e h.

Abaixo definimos os tipos de ataque a Esquemas de Assinaturas Digitais e em seguida os objetivos de um adversário, conforme inicialmente estabelecidos por Goldwasser *et al.* [32].

- **Tipos de ataque a Esquemas de Assinaturas Digitais:**

Ataque de chave pública (*Key-Only attack*) - O adversário tem acesso apenas a chave pública.

Ataque de mensagem conhecida (*Known-message attack*) - O adversário tem acesso a um determinado número de pares mensagem/assinatura legítimos.

Ataque de mensagem escolhida (*Chosen-message attack*) - O adversário escolhe um determinado número de mensagens e tem acesso a suas respectivas assinaturas legítimas.

Ataque adaptativo de mensagem escolhida (*Adaptive chosen-message attack*) - O adversário tem acesso ao assinante como um oráculo, podendo requerer assinatura de mensagens escolhidas de acordo com os resultados que vai obtendo.

- **Objetivos do Adversário:**

Falsificação existencial (*Existential Forgery*) - Quando é possível gerar uma assinatura válida para uma mensagem, mas sem conseguir controlar o conteúdo da mensagem.

Falsificação selecionada (*Selective Forgery*) - Adversário consegue criar uma assinatura válida para uma determinada mensagem ou classe de mensagens previamente escolhida.

Falsificação universal (*Universal Forgery*) - É possível falsificar qualquer mensagem utilizando-se apenas as informações públicas disponíveis.

Quebra total (*Total Break*) - Adversário consegue reconstruir a chave privada a partir das informações públicas disponíveis.

2.8 Paradigma Fiat-Shamir

2.8.1 Transformação de Esquemas de Identificação em Assinatura

Em [28], Amos Fiat e Adi Shamir propuseram um esquema de identificação baseado na dificuldade de fatoração de inteiros e, principalmente, mostraram como construir um esquema de assinatura a partir daquele esquema de identificação, originando assim o conhecido paradigma Fiat-Shamir, também referenciado como transformação Fiat-Shamir.

Pointcheval e Stern [51, 52] proveram argumentos de segurança no modelo do oráculo aleatório que podem ser utilizados para qualquer esquema de assinatura derivado de um esquema de identificação seguro com a utilização do paradigma Fiat-Shamir. Posteriormente, Abdalla *et al.* [1] especificaram mais claramente o que seria um esquema de identificação seguro, neste contexto, e quais seriam as condições minimais, no sentido de necessário e suficiente, do esquema de identificação para garantir a segurança do esquema de assinatura gerado, considerando novamente o modelo do oráculo aleatório.

Para apresentarmos um exemplo simplificado da utilização de um esquema de identificação $\mathcal{ID}$ como esquema de assinatura digital $\mathcal{AS}$, podemos utilizar os algoritmos Setup e Gen de $\mathcal{ID}$ sem alteração em $\mathcal{AS}$ e criamos os algoritmos Assina e Verifica de $\mathcal{AS}$ conforme o pseudocódigo abaixo.

Pseudocódigo dos algoritmos do esquema de assinatura

Assina(sk, msg) {	**Verifica**(pk, msg, σ) {
Cmt $\leftarrow$ P(sk)	interpreta σ como $Cmt \| Rsp$
Ch $\leftarrow$ Hash($Cmt \| msg$)	Ch $\leftarrow$ Hash($Cmt \| msg$)
Rsp $\leftarrow$ P(sk,$Cmt \| Ch$)	Dec $\leftarrow$ V($pk, Cmt \| Ch \| Rsp$)
retorna $Cmt \| Rsp$	retorna Dec
}	}

Capítulo 3

Criptossistemas de Chave Pública Multivariável

Como vimos, uma das motivações para a nossa pesquisa é o risco do comprometimento dos atuais sistemas criptográficos de chave pública no caso do desenvolvimento de computadores quânticos com a capacidade adequada. Dentre as classes já citadas como pós-quânticas, vamos agora explorar os Criptossistemas de Chave Pública Multivariável (MPKC).

3.1 Problema $\mathcal{MQ}$

Seja $\mathbb{F}$ um Corpo Finito de Galois de ordem q, $n \in \mathbb{N}$ o número de variáveis, $m \in \mathbb{N}$ o número de equações, $d \in \mathbb{N}$ o grau do sistema de equações, $p(x_1, ..., x_n)$ um polinômio de grau d em n variáveis sobre $\mathbb{F}$ e $\mathcal{P} = (p_1, \ldots, p_m)$ um sistema de m polinômios de grau d em n variáveis sobre $\mathbb{F}$. O problema do Sistema de Equações Polinomiais Multivariáveis Simultâneas consiste em encontrar $\mathrm{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ tal que $\mathcal{P}(\mathrm{x}) = \mathrm{y}$, sendo $\mathrm{y} = (y_1, \ldots, y_m) \in \mathbb{F}^m$:

$$\mathcal{P} = \begin{cases} p_1(x_1, ..., x_n) = y_1 \\ p_2(x_1, ..., x_n) = y_2 \\ \vdots \\ p_m(x_1, ..., x_n) = y_m \end{cases}$$

Quando o grau do sistema de equações $\mathcal{P}$ é igual a 2 ($d = 2$), chamamos então de Problema $\mathcal{MQ}$. Patarin e Goubin [47] demonstraram que o problema $\mathcal{MQ}$ é NP-Completo, não se conhecendo até hoje nenhum algoritmo, nem mesmo quântico, de tempo polinomial que possa resolver este problema [5], contribuindo assim para a segurança de sistemas criptográficos baseados nesta primitiva. Infelizmente, os MPKC atuais não tem sua segurança baseada exclusivamente no problema $\mathcal{MQ}$, visto que não sabemos como inserir uma trapdoor em uma instância aleatória de função $\mathcal{MQ}$, sendo então necessária uma

construção específica, como veremos mais a frente, que resulta na dependência de outros problemas, sendo mais frequente o problema do Isomorfismo de Polinômios (IP).

Chamamos o Sistema de Equações Polinomiais Multivariáveis Quadráticas $\mathcal{P}$ de Função $\mathcal{MQ}$ e a denotamos por $\mathcal{MQ}(n, m, \mathbb{F})$ para estabelecer suas propriedades. Abaixo vemos o formato genérico da família de funções $\mathcal{MQ}$.

$$\begin{aligned} p_1(x_1, ..., x_n) &:= \sum_{1 \leq i \leq j \leq n} a_{1,i,j} x_i x_j + \sum_{i=1}^{n} b_{1,i} x_i + \delta_1 \\ &\vdots \\ p_l(x_1, ..., x_n) &:= \sum_{1 \leq i \leq j \leq n} a_{l,i,j} x_i x_j + \sum_{i=1}^{n} b_{l,i} x_i + \delta_l \\ &\vdots \\ p_m(x_1, ..., x_n) &:= \sum_{1 \leq i \leq j \leq n} a_{m,i,j} x_i x_j + \sum_{i=1}^{n} b_{m,i} x_i + \delta_m \end{aligned}$$

Precisamos destacar que ao estabelecer os parâmetros para instanciação de uma função $\mathcal{MQ}$ devemos ter $n >= m$, visto que no caso contrário ($n < m$) teríamos um sistema superdefinido, onde existiriam mais equações do que variáveis, o que tornaria "fácil" a solução.

Os MPKC foram de alguma forma originados do trabalho de Matsumoto e Imai [41] e talvez não tivessem sobrevivido sem as contribuições de Patarin, conforme destacado por Stern [61]. O fato de serem considerados como alternativa pós-quântica é talvez a menor das vantagens. Os MPKC possibilitam aplicações em esquemas de assinatura digital com tamanhos reduzidos de assinatura (*short signatures*) [13], as implementações dos esquemas tem se mostrado extremamente rápidos e eficientes, tanto em software [11] quanto em hardware, e eles são indicados como uma opção para sistemas embarcados com restrições de processamento. Teríamos ainda muitas outras razões a apontar, mas é desnecessário visto que já podemos concluir que os MPKC podem vir a ser uma alternativa para esquemas de chave pública convencionais, baseados em RSA e ECC, e por isso devemos prosseguir as pesquisas que vem sendo realizadas neste sentido.

3.2 Tamanho da chave pública MPKC

Em um sistema de equações polinomiais multivariáveis de grau d, o número total de termos em um polinômio é igual a:

$$\binom{n+d}{d} = \frac{(n+d)!}{n! \cdot d!}$$

Dessa forma podemos ver que o tamanho da chave pública $\mathcal{P}$ cresce em $O(mn^d)$, por isso a grande maioria dos criptossistemas é projetado com $d = 2$, ou seja, quadrático. Para

exemplificar esse crescimento, apresentamos na tabela 3.1 o número de termos em um polinômio e também o total de termos para um sistema com 84 variáveis e 80 equações quando $d \in \{2, 3, 4\}$.

Considerando $n = 84, m = 80$	$d = 2$	$d = 3$	$d = 4$
nº de termos em 1 polinômio	3.655	105.995	2.331.890
nº total de termos do sistema	292.400	8.479.600	186.551.200

Tabela 3.1: *Exemplo do crescimento do número total de termos de um sistema de equações polinomiais multivariáveis em razão do grau d*

É fácil notar que o tamanho da chave pública é extremamente sensível ao crescimento de d. Além disso, qualquer sistema polinomial de grau $d > 2$ pode ser re-escrito como um sistema de grau 2 acrescentando-se mais variáveis, o que sugere que o aumento de d não necessariamente aumenta a segurança do sistema, mas certamente irá prejudicar sua eficiência com tamanhos de chaves muito grandes. Dessa forma, vamos estudar os casos em que $d = 2$.

Para calcularmos o nº de termos de um polinômio multivariável quadrático p_l, considerando a existência de um termo constante, utilizamos a seguinte fórmula:

$$\tau(n) := \begin{cases} 1 + n + \frac{n(n-1)}{2} = 1 + \frac{n(n+1)}{2} & \text{para } \mathbb{F} = GF(2) \\ 1 + n + \frac{n(n+1)}{2} = 1 + \frac{n(n+3)}{2} & \text{para demais casos} \end{cases}$$

Assim, o tamanho total da chave pública $\mathcal{P}$ é igual a $m \cdot \tau(n) \cdot \psi(\mathbb{F})$, onde $\psi(\mathbb{F})$ é a quantidade de bits necessária para representar um elemento de $\mathbb{F}$

3.3 Visão geral dos MPKC

A função $\mathcal{MQ}$ é uma função unidirecional [47], porém, como sabemos, para construção de um criptossistema de chave pública é necessário possuirmos uma função unidirecional com segredo (*trapdoor*), o que simplesmente não é possível alcançar a partir de uma instância aleatória de função $\mathcal{MQ}$, como destacam de forma simples e direta Ding e Yang em [5, p. 194]. Por isso, os atuais MPKC contém um mapeamento $F : \mathbb{F}^n \mapsto \mathbb{F}^m$, inversível, utilizado juntamente com duas transformações afins S e T para criação de uma função $\mathcal{MQ}$. Isto é, eles são construídos a partir de um conjunto de equações polinomiais "fáceis", utilizando-se duas transformações afins para criar uma instância aparentemente aleatória de uma função $\mathcal{MQ}$, que servirá de chave pública. Assim, a chave pública $\mathcal{P}$, que é uma função $\mathcal{MQ}$, é criada a partir da composição de duas transformações afins $S : \mathbb{F}^n \mapsto \mathbb{F}^n, T : \mathbb{F}^m \mapsto \mathbb{F}^m$ e um mapeamento central $F : \mathbb{F}^n \mapsto \mathbb{F}^m$, ou seja, $\mathcal{P}(x) = T \circ F \circ S = T(F(S(x)))$, sendo T, F e S a chave privada. Na figura 3.1 vemos a representação gráfica de uma $\mathcal{MQ}$-Trapdoor, conforme acabamos de explicitar.

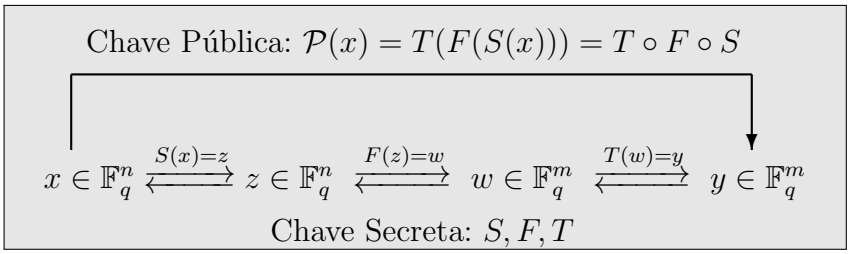

Figura 3.1: *Representação gráfica de uma $\mathcal{MQ}$-Trapdoor*

Como dissemos no final da seção 3.1, uma característica dos MPKC conhecidos atualmente é que nenhum tem sua segurança baseada exclusivamente no problema $\mathcal{MQ}$, sendo a grande maioria dependente também do problema de Isomorfismo de Polinômios (IP), o qual podemos dizer de forma simplificada que, consiste em, dados $\mathcal{P}$ e F, encontrar duas transformações afins S e T tais que $T \circ F \circ S = T(F(S(x)))) = \mathcal{P}(x)$, sendo $\mathcal{P}$, uma função $\mathcal{MQ}$ utilizada como chave pública de um MPKC e F um mapeamento inversível utilizado juntamente com S e T como chave secreta do mesmo criptossistema. Acredita-se que uma instância aleatória do problema IP seja difícil, porém, quando aplicado às atuais $\mathcal{MQ}$-Trapdoor ainda suscita controvérsia quanto à sua intratabilidade, já tendo inclusive possibilitado a quebra de vários outros esquemas e protocolos criptográficos anteriormente apresentados, como o SFlash [7, 15, 22, 48] que chegou a ser recomendado pelo NESSIE. Esta vulnerabilidade existe pois as atuais $\mathcal{MQ}$-Trapdoor possuem esta estrutura específica de construção que consiste na composição de um mapeamento central F de formato determinado, com duas transformações afins (S e T), resultando em uma forma limitada de função $\mathcal{MQ}$, e que, em consequência, possibilita em alguns casos soluções fáceis desta versão do problema IP.

Apresentaremos mais a frente a relação das $\mathcal{MQ}$-Trapdoor conhecidas atualmente, mas já temos aqui uma visão geral do modelo atual utilizado nos MPKC, o qual podemos ver sintetizado na Figura 3.2, extraída da apresentação de Sakumoto na CRYPTO'2011.

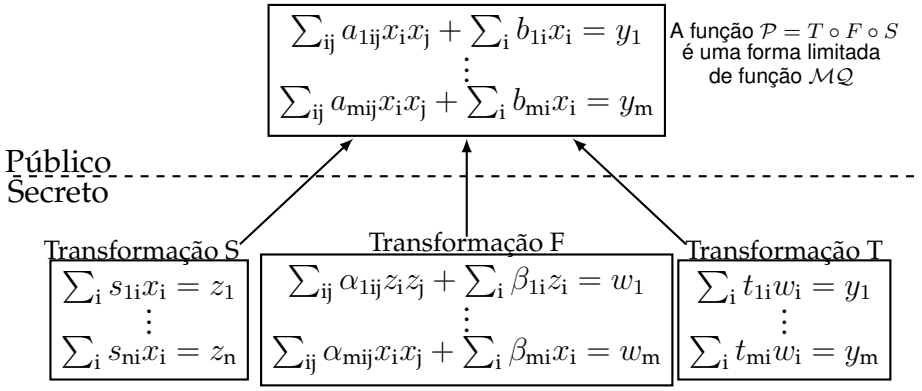

Figura 3.2: *Atual Modelo MPKC*

3.4 Resumo genérico de operações em MPKC

Abaixo vamos especificar o funcionamento genérico das operações de cifração, decifração, assinatura e verificação. Nossa intenção não é esgotar todos os detalhes atinentes a cada operação de algum criptossistema específico, pelo contrário, queremos explicitar o funcionamento comum a todos os MPKC registrando assim os conhecimentos necessários para buscarmos a consecução de nossos objetivos.

3.4.1 Cifração

Para cifrarmos um texto claro x utilizamos, como em qualquer esquema de criptografia assimétrica, uma chave pública $\mathcal{P}$. No caso dos MPKC teríamos então um novo texto cifrado $y \leftarrow \mathcal{P}(x)$, porém isso apenas não é suficiente, pois dependendo do mapeamento central F utilizado, dado um valor $w \in \mathbb{F}^m$ é possível que existam várias pré-imagens deste valor em $\mathbb{F}^n$. Para que seja possível então durante a decifração garantir que foi encontrada a pré-imagem correta e consequentemente o texto claro original, é efetuada uma computação paralela de um valor de hash de x, ou seja, fazemos $\bar{x} \leftarrow H(x)$ e o texto cifrado passa a ser composto pelo par $(y, \bar{x}) \in \mathbb{F}^m \times \{0,1\}^l$, sendo l o tamanho da saída da função de hash H.

3.4.2 Decifração

Dado um texto cifrado $(y, \bar{x})$, precisamos da chave secreta composta por T, F e S para efetuarmos a decifração. Pode ser necessário encontrar todas as raízes possíveis da equação, ou seja, buscar todas as pré-imagens existentes de y, o que pode constituir uma considerável sobrecarga que dificulta a construção de esquemas eficientes baseados neste modelo clássico MPKC. Efetuamos então a decifração com:

$$y \in \mathbb{F}^m \xrightarrow{T^{-1}(y)=w} w \in \mathbb{F}^m \xrightarrow{F^{-1}(w)=z} z \in \mathbb{F}^n \xrightarrow{S^{-1}(z)=x'} x' \in \mathbb{F}^n$$

Como dissemos logo acima, o valor de hash $\bar{x}$ é utilizado então para verificarmos se foi encontrada a pré-imagem correta, ou seja, o texto claro original.

$$H(x') \stackrel{?}{=} \bar{x}$$

Se o teste falhar, repetimos a operação buscando uma outra solução (pré-imagem) até que seja encontrada a correta.

3.4.3 Assinatura

A realização de uma assinatura é semelhante ao processo de decifração, mas sem a necessidade de buscar uma pré-imagem específica, ou seja, calculamos apena uma vez podendo utilizar qualquer solução visto que todas irão satisfazer o processo de verificação desta assinatura. Assim, dado um texto claro y precisamos efetuar apenas uma vez o processo a seguir para gerarmos uma assinatura σ:

$$y \in \mathbb{F}^m \xrightarrow{T^{-1}(y)=w} w \in \mathbb{F}^m \xrightarrow{F^{-1}(w)=z} z \in \mathbb{F}^n \xrightarrow{S^{-1}(z)=\sigma} \sigma \in \mathbb{F}^n$$

3.4.4 Verificação de Assinatura

A verificação de assinatura é o procedimento mais simples de explicarmos e também de implementarmos. Dada uma assinatura $\sigma \in \mathbb{F}^n$ gerada com a chave privada de $\mathcal{A}lice$, para efetuarmos a verificação vamos utilizar a chave pública $\mathcal{P}$ de $\mathcal{A}lice$ e comparar com a mensagem original $y \in \mathbb{F}^m$, sendo aceita a assinatura se o teste for bem sucedido.

$$\mathcal{P}(\sigma) \stackrel{?}{=} y$$

3.5 $\mathcal{MQ}$-Trapdoors

Na Seção 3.3 vimos que todo MPKC constrói a sua chave pública, que é uma função $\mathcal{MQ}$, a partir da composição de um mapeamento central F e duas transformações afins S e T. O formato desse mapeamento central F é uma das principais diferenças entre as $\mathcal{MQ}$-trapdoor , porém, como não utilizaremos nenhuma delas para o desenvolvimento de nossos objetivos, nos permitimos manter o detalhamento dos mapeamentos centrais e demais detalhes técnicos específicos fora do escopo desta dissertação, visando assim privilegiar a fluidez da leitura, sem comprometer o entendimento dos assuntos necessários. Outra grande diferença que precisamos destacar é que alguns esquemas utilizam, além do corpo $\mathbb{F}$ um corpo extensão $\mathbb{E}$. Chamamos esses esquemas de "Big Field", sendo os demais nomeados de "Single Field".

Atualmente são conhecidas cinco $\mathcal{MQ}$-trapdoors básicas a partir das quais se originam inúmeros esquemas MPKC. Na tabela 3.2 listamos essas $\mathcal{MQ}$-trapdoors básicas relacionando seus autores, ano de criação e respectivas referências para consulta a mais detalhes sobre as mesmas. Em suas versões básicas todas as $\mathcal{MQ}$-trapdoors são consideradas quebradas, necessitando que sejam utilizados modificadores genéricos para que sejam seguros, como veremos a seguir.

Nome (Acrônimo)	Autores	Ano	Ref
Matsumoto Imai Scheme A (MIA)	Matsumoto e Imai	1988	[41]
Hidden Field Equations (HFE)	Patarin	1996	[46]
Unbalanced Oil and Vinegar (UOV)	Kipnis, Patarin, e Goubin	1999	[37]
Stepwise Triangular Systems (STS)	Wolf, Braeken, e Preneel	2004	[67]
ℓ-Invertible Cycles (ℓ-IC)	Ding, Wolf, e Yang	2007	[21]

Tabela 3.2: *Listagem das $\mathcal{MQ}$-trapdoors básicas conhecidas atualmente*

3.6 Modificadores Genéricos

No decorrer das pesquisas e do desenvolvimento dos MPKC, assistimos a um ciclo consistindo na apresentação de um novo esquema, que em seguida é quebrado e então prepara-se um "conserto", dando origem a um novo esquema e ao recomeço do ciclo. A maioria desses "consertos" acabam podendo ser generalizados e aplicados a diversas $\mathcal{MQ}$-Trapdoor diferentes. Wolf e Preneel em sua Taxonomia dos Esquemas de Chave Pública Multivariáveis Quadráticas [66], bem destacaram a possibilidade e o benefício da utilização de modificadores genéricos, ressaltando que dependendo da $\mathcal{MQ}$-Trapdoor algumas opções podem se mostrar mais eficientes do que outras.

Na tabela 3.3 sintetizamos a situação dos modificadores genéricos conhecidos:

	Simbolo e Nome	Segurança	Idéia Básica
-	Minus	seguro	descarta alguns polinômios
+	Plus	maioria sem efeito	adiciona polinômios
p	Prefix ou Postfix	em aberto	força alguns $w_l = 0$
v	Vinegar	pouco mais seguro	pertubation em pequeno subespaço
i	Internal Pertubation	em aberto	igual a p+v
f	Fixing	em aberto	algumas variáveis aleatórias
m	Masking	em aberto	descarta algumas variáveis
s	Sparse	em aberto	usa polinômios esparsos

Tabela 3.3: *Listagem de Modificadores Genéricos para MPKC*

3.7 Novo Modelo MPKC de Sakumoto *et al*

Sakumoto, Shirai e Hiwatari [55] apresentaram durante a CRYPTO'2011 dois novos protocolos de identificação de conhecimento-zero os quais chamamos de $\mathcal{MQID}$-3 e $\mathcal{MQID}$-5. Estes protocolos se baseiam em um novo modelo de MPKC, o qual tem sua segurança reduzida ao problema $\mathcal{MQ}$. No próximo Capítulo veremos mais detalhes sobre os protocolos propriamente, mas por agora vamos resumir esse novo e interessante modelo de MPKC que possui uma distinta e importante característica, a utilização de uma instância aleatória de função $\mathcal{MQ}$, ao invés da forma limitada descrita anteriormente. Neste novo modelo uma chave privada s passa a ser os valores das n variáveis nas m equações que compõe a função $\mathcal{MQ}$, isto é, s $= (x_1, ..., x_n) \in \mathbb{F}^n$, sendo a chave pública então o resultado v $= (y_1, ..., y_m) \in \mathbb{F}^m$ e a função $\mathcal{MQ}$ em si passa a ser uma instância aleatória disponível como parâmetro do sistema para todos os usuários, conforme vemos resumido na figura 3.3.

Obviamente, por se tratar de uma instância aleatória de função $\mathcal{MQ}$, não se consegue construir uma trapdoor a partir dela, como destacam Ding e Yang de forma simples e direta em [5, p. 194]. Mas, em compensação, neste modelo o tamanho das chaves se torna

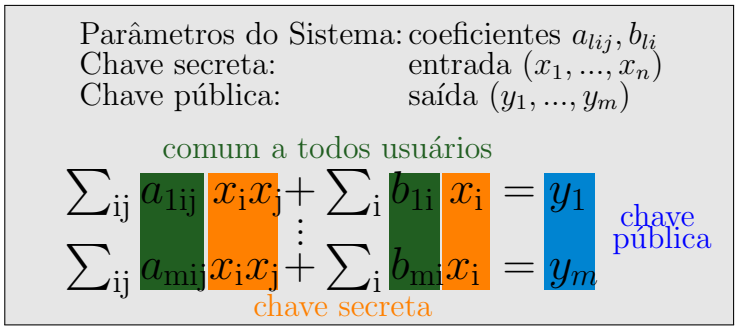

Figura 3.3: *Novo Modelo MPKC de Sakumoto-Shirai-Hiwatari*

extremamente pequeno, sendo praticamente igual ao parâmetro de segurança desejado, ou seja, para termos uma segurança de κ bits, seria suficiente um par de chaves de apenas κ bits, conforme detalharemos mais a frente. Uma outra consequência direta disto é que a geração de chaves se torna extremamente simples e eficiente, bastando sortear uma chave privada qualquer e executar uma única vez a função $\mathcal{MQ}$, que é o parâmetro do sistema, para chegar a respectiva chave pública. Finalmente, devemos ressaltar o fato que neste modelo, com utilização de apenas instâncias aleatórias de função $\mathcal{MQ}$, foi eliminada qualquer dependência da sua segurança com outros problemas que não o próprio problema $\mathcal{MQ}$, sendo essa talvez uma de suas mais importantes características.

3.8 Criptanálise do Problema $\mathcal{MQ}$ e dos MPKC

No caso de Esquemas de Identificação o objetivo principal de um adversário $\mathcal{C}arlos$ é conseguir se passar por uma provadora $\mathcal{A}lice$ perante um verificador $\mathcal{B}eto$. Para tanto, $\mathcal{C}arlos$ poderia tentar formular a chave privada de $\mathcal{A}lice$, e se obtivesse sucesso, com certeza poderia alcançar seu objetivo principal. Vemos que pelo modelo Sakumoto-Shirai-Hiwatari exposto na seção anterior, para $\mathcal{B}eto$ extrair a chave privada de $\mathcal{A}lice$, mesmo que esteja disponível uma forma de realizar um ataque ativo simultâneo, ou seja, onde o adversário $\mathcal{C}arlos$ age como verificador realizando diversas validações de identidade simultâneas de $\mathcal{A}lice$, ainda assim seria necessário que ele resolvesse um problema $\mathcal{MQ}$, encontrando então um valor s' $\in \mathbb{F}^n$ tal que $\mathcal{P}$(s') =v, sendo v $\leftarrow \mathcal{P}$(s) a chave pública de $\mathcal{A}lice$ e s $\in \mathbb{F}^n$ a sua chave secreta.

Vale ressaltar que as técnicas de criptanálise que iremos ver foram desenhadas para atacar os MPKC existentes com objetivo de decifrar um determinado texto cifrado y resolvendo um sistema polinomial tal que $\mathcal{P}(x) - y = 0$. No contexto do modelo de Sakumoto *et al.* esse mesmo ataque pode ser utilizado então para recuperar uma chave secreta a partir da chave pública respectiva, como descrito no parágrafo anterior.

Existem diversos ataques especializados em determinadas $\mathcal{MQ}$-Trapdoor, que utilizam-se das características específicas das construções dos mapeamentos centrais para alcançar uma vantagem ou até mesmo quebrar um criptossistema com determinada escolha de

parâmetros. Esses ataques não são de nosso interesse visto que não se aplicam ao modelo Sakumoto-Shirai-Hiwatari por este utilizar uma instância aleatória de função $\mathcal{MQ}$. Apenas como referência podemos citar os bem sucedidos ataques ao SFLASH, formulados primeiramente por Dubois, Fouque, Shamir e Stern em 2007 [22], quebrando os parâmetros originais e mais relevantes do SFLASH, e a posterior proposta de Bouillaguet *et al.* [7] que possibilitou a quebra para qualquer escolha de parâmetros.

Feldmann [27] dividiu os ataques apresentados em seu trabalho em dois grupos, o primeiro contendo apenas ataques que procuram resolver um problema $\mathcal{MQ}$ sem utilizar nenhuma informação além da própria chave pública do sistema, e o segundo grupo que é composto por ataques mais especializados e que tentam obter vantagem a partir do conhecimento da estrutura específica de construção de algum determinado esquema. Como já dissemos, para o nosso trabalho interessam os ataques do primeiro grupo, visto que os demais não se aplicam ao modelo Sakumoto-Shirai-Hiwatari por este utilizar uma instância aleatória de função $\mathcal{MQ}$.

Praticamente todos os ataques genéricos conhecidos hoje em dia utilizam de alguma forma bases de Gröbner, teoria formulada em 1965 por Buchberger [8]. Durante algum tempo, mais precisamente de 2000 até 2004, o algoritmo XL (eXtended Linearization) foi considerado como a principal proposta para solução de um sistema $\mathcal{MQ}$, tendo sido apresentado originalmente por Courtois *et al.* [12] na EuroCrypt'2000. Entretanto, Ars *et al.* [3] demonstraram em 2004 que na verdade, o algoritmo XL nada mais é do que uma versão mais lenta do algoritmo F4 de Faugère [24], sendo então na verdade uma variante da base de Gröbner, não constituindo uma alternativa, mas sim sendo apenas uma especialização deste. Outro que citamos aqui apenas por uma questão histórica e por ser muito semelhante ao XL é o Zhuang-Zi, proposto por Ding *et al.* [20], e que apenas difere do algoritmo XL no seu passo 3 quando efetua a escolha de um polinômio de menor grau ao invés do formato definido no XL. Os próprios autores do Zhuang-Zi nas conclusões de sua proposta já alertavam que não era esperado que aquele algoritmo suplantasse o estado-da-arte – F4 e F5.

O estado-da-arte atual para resolver o problema $\mathcal{MQ}$ em $\mathbb{F}_2$ foi apresentado em 2010 por Bouillaguet, Chen, Cheng, Chou, Niederhagen, Shamir, e Yang [6]. Ao invés de utilizarem base de Gröbner, principalmente devido a necessidade exponencial de memória para sua implementação, Bouillaguet *et al.* resolveram utilizar uma abordagem diferente, baseando-se no algoritmo padrão de busca exaustiva. Eles demonstraram como encontrar todos os zeros de um polinômio de grau d com n variáveis em apenas $d \cdot 2^n$ operações binárias. À partir deste resultado, eles adaptaram a mesma técnica utilizada em um único polinômio para encontrar todos os zeros em comum de um conjunto de m polinômios quadráticos aleatórios em $\lg n \cdot 2^{n+2}$ operações binárias.

Entendemos que não acrescentaria aos nossos objetivos uma explanação maior sobre as técnicas aqui citadas, podendo todos os detalhes serem consultados em seus artigos

originais constantes das referências. Nossa intenção foi registrar os resultados alcançados até o momento na criptanálise MPKC para que possamos escolher parâmetros de segurança adequados e mantermos assim o escopo deste trabalho focado em nosso objetivo principal.

Capítulo 4

Aprimorando o protocolo $\mathcal{MQID}$-3

Neste capítulo vamos apresentar um novo protocolo de identificação canônico baseado no $\mathcal{MQID}$-3 de Sakumoto-Shirai-Hiwatari, porém com probabilidade de personificação 1/2 em uma rodada, ao invés da probabilidade de 2/3 do protocolo original, conseguindo assim a diminuição do número de rodadas necessárias para o mesmo nível de segurança e diminuindo o tamanho da comunicação necessária em uma execução. Demonstraremos que nosso protocolo aprimorado tem uma comunicação 9,5% menor que a original para o mesmo nível de segurança, considerando os parâmetros propostos inicialmente e a já citada utilização da técnica sugerida por Jacques Stern [60] para reduzir o número de compromissos trafegados, enviando apenas um hash de todos os compromissos no primeiro passo do protocolo e depois incluindo na resposta do provador apenas aqueles que não podem ser reconstituídos, para que juntamente com os compromissos validados pelo verificador seja possível efetuar um novo hash para comparar com o recebido inicialmente. Começaremos fazendo uma revisão do Esquema de Identificação de Sakumoto-Shirai-Hiwatari apresentado na CRYPTO'2011 [55] e então introduziremos nossa proposta com as necessárias análises.

4.1 Revisão do Esquema de Identificação $\mathcal{MQID}$

No esquema de identificação de Sakumoto-Shirai-Hiwatari o algoritmo Setup recebe um parâmetro de segurança 1^κ que determina n, m e $\mathbb{F}$ e retorna um parâmetro de sistema $\mathcal{P} \in_R \mathcal{MQ}(n, m, \mathbb{F})$, o qual é uma instância aleatória de função $\mathcal{MQ}$. O algoritmo Gen sorteia um valor aleatório s $\in_R \mathbb{F}^n$ que será a chave privada, e executa $\mathcal{P}$(s) =v, sendo v então a chave pública.

O protocolo $\mathcal{MQID}$-3 utiliza uma função bilinear $\mathcal{G}$, que chamamos de forma polar de $\mathcal{P}$, sendo $\mathcal{G}(a, b) = \mathcal{P}(a + b) - \mathcal{P}(a) - \mathcal{P}(b)$, com $a, b \in \mathbb{F}^n$.

Os autores utilizaram uma abordagem de divisão-escolha, onde o segredo é dividido em vários pedaços e o verificador escolhe algum para validar. A ideia básica é que o provador

demonstre que possui uma tupla $(r_0, r_1, t_0, t_1, e_0, e_1)$ que satisfaça:

$$\mathcal{G}(t_0, r_1) + e_0 = \mathcal{P}(\mathrm{s}) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1 \tag{4.1}$$

$$\text{e} \qquad (t_0, e_0) = (r_0 - t_1, \mathcal{P}(r_0) - e_1) \tag{4.2}$$

A equação (4.1) acima vem da seguinte maneira, utilizando-se a bilinearidade de $\mathcal{G}$:

$$\left.\begin{aligned} \mathcal{G}(r_0, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\ \mathcal{G}(t_0, r_1) + \mathcal{G}(t_1, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\ \mathcal{G}(t_0, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) \\ \mathcal{G}(t_0, r_1) &= \mathcal{P}(s) - (e_0 + e_1) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) \\ \mathcal{G}(t_0, r_1) + e_0 &= \mathcal{P}(s) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1 \end{aligned}\right\} \tag{4.3}$$

Para isso a chave secreta s é dividida da forma descrita a seguir. Inicialmente são escolhidos $r_0, t_0 \in_R \mathbb{F}^n$ e $e_0 \in_R \mathbb{F}^m$, e então calculados $r_1 \leftarrow \mathrm{s} - r_0$, $t_1 \leftarrow r_0 - t_0$ e $e_1 \leftarrow \mathcal{P}(r_0) - e_0$, ou seja, $\mathrm{s} = r_0 + r_1 = r_1 + t_0 + t_1$ e $\mathcal{P}(r_0) = e_0 + e_1$. Na Figura 4.1 podemos ver uma representação gráfica dessa divisão. Vale ressaltar que nenhuma informação da chave secreta pode ser recuperada possuindo-se apenas duas das três partições finais.

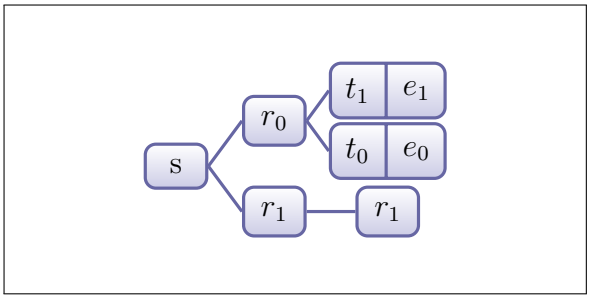

Figura 4.1: *Representação gráfica da divisão do segredo no protocolo $\mathcal{MQID}$-3*

O protocolo $\mathcal{MQID}$-3 possui probabilidade de personificação de 2/3 em uma rodada. Por isso, para termos uma probabilidade total de personificação menor que $2^{-\lambda}$, é preciso repetir o protocolo um total de $\lceil \frac{\lambda}{\lg 3 - 1} \rceil$ vezes, como podemos ver abaixo, onde r é o número total de rodadas necessárias:

$$\left(\frac{2}{3}\right)^r \leq 2^{-\lambda}$$

$$r(\lg 2 - \lg 3) \leq -\lambda \cdot \lg 2$$

$$r(1 - \lg 3) \leq -\lambda$$

$$r(\lg 3 - 1) \geq \lambda$$

$$r \geq \frac{\lambda}{\lg 3 - 1}$$

A comunicação em uma rodada é igual a $2n+5m+2$ bits, visto que o provador gera um

hash hc dos 3 compromissos criados, o qual possui $2m$ bits, e o envia para o verificador, que responde com um desafio Ch de tamanho 2 bits, do qual decorre o envio de Rsp pelo provador, o qual possui $2n + 3m$ bits. Vemos então que a comunicação total seria de $r \cdot (2n + 5m + 2)$ bits, ou $52 \cdot (2 \cdot 84 + 5 \cdot 80 + 2) = 29640$ bits, com os parâmetros originalmente propostos.

A Figura 4.2 apresenta o protocolo $\mathcal{MQID}$-3 original.

Entrada do Provador: $((\mathcal{P}, \mathrm{v}), \mathrm{s})$		Entrada do Verificador: $(\mathcal{P}, \mathrm{v})$
Escolhe $r_0, t_0 \in_R \mathbb{F}^n$, $e_0 \in_R \mathbb{F}^m$		
$r_1 \leftarrow \mathrm{s} - r_0$, $t_1 \leftarrow r_0 - t_0$		
$e_1 \leftarrow \mathcal{P}(r_0) - e_0$		
$c_0 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$		
$c_1 \leftarrow Com(t_0, e_0)$		
$c_2 \leftarrow Com(t_1, e_1)$		
$hc \leftarrow H(c_0, c_1, c_2)$		
	$\xrightarrow{hc}$	
		Escolhe $Ch \in_R \{0, 1, 2\}$
	$\xleftarrow{Ch}$	
Se $Ch = 0, Rsp \leftarrow (r_0, t_1, e_1, c_0)$		
Se $Ch = 1, Rsp \leftarrow (r_1, t_1, e_1, c_1)$		
Se $Ch = 2, Rsp \leftarrow (r_1, t_0, e_0, c_2)$		
	$\xrightarrow{Rsp}$	
		Se $Ch = 0$, utiliza $Rsp = (r_0, t_1, e_1, c_0)$ para verificar:
		$c'_1 \leftarrow Com(r_0 - t_1, \mathcal{P}(r_0) - e_1)$
		$c'_2 \leftarrow Com(t_1, e_1)$
		$hc \stackrel{?}{=} H(c_0, c'_1, c'_2)$
		Se $Ch = 1$, utiliza $Rsp = (r_1, t_1, e_1, c_1)$ para verificar:
		$c'_0 \leftarrow Com(r_1, \mathrm{v} - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1)$
		$c'_2 \leftarrow Com(t_1, e_1)$
		$hc \stackrel{?}{=} H(c'_0, c_1, c'_2)$
		Se $Ch = 2$, utiliza $Rsp = (r_1, t_0, e_0, c_2)$ para verificar:
		$c'_0 \leftarrow Com(r_0, \mathcal{G}(t_0, r_1) + e_0)$
		$c'_1 \leftarrow Com(t_0, e_0)$
		$hc \stackrel{?}{=} H(c'_0, c'_1, c_2)$

Figura 4.2: *Resumo do protocolo* $\mathcal{MQID}$-3

4.2 $\mathcal{MQID}$-3 Aprimorado

Primeiramente, vale destacar que em [55] também foi introduzido o protocolo $\mathcal{MQID}$-5, o qual já possui probabilidade de personificação 1/2 em uma rodada. Porém, ele não é canônico [1], o que impossibilitaria a geração a partir dele de um esquema de assinaturas pelo paradigma da transformação Fiat-Shamir [28]. Muito recentemente, Alaoui *et al.* [2] estendeu o Forking Lemma de Pointcheval e Stern [51, 52] e a transformação Fiat-Shamir

para possibilitar a utilização de protocolos com $2n + 1$ passos, utilizando exatamente o protocolo $\mathcal{MQID}$-5 de Sakumoto, Shirai e Hiwatari como exemplo, porém essa assinatura seria teoricamente mais lenta que uma baseada na nossa proposta, visto que a solução derivada do protocolo em 5 passos necessita de $2r$ hashes para simular o verificador, enquanto a nossa utilizaria apenas um hash.

Para conseguirmos aprimorar o protocolo de identificação $\mathcal{MQID}$-3, incrementamos a divisão do segredo de forma a possibilitar duas alternativas exclusivas de verificação. Vimos que no protocolo original a chave privada s é dividida em r_0 e r_1, sendo s$= r_0 + r_1$. E que depois é efetuada a divisão de r_0 em t_0 e t_1 e de $\mathcal{P}(r_0)$ em e_0 e e_1, da mesma forma. Nós mantemos as mesmas divisões originais e acrescentamos a divisão de r_1 em d_0 e d_1 e de $\mathcal{P}(r_1)$ em u_0 e u_1, efetuando o seguinte: escolhemos $d_0 \in_R \mathbb{F}^n$ e $u_0 \in_R \mathbb{F}^m$, e calculamos $d_1 \leftarrow r_1 - d_0$ e $u_1 \leftarrow F(r_1) - u_0$. Visualizamos essa nova divisão na figura 4.3.

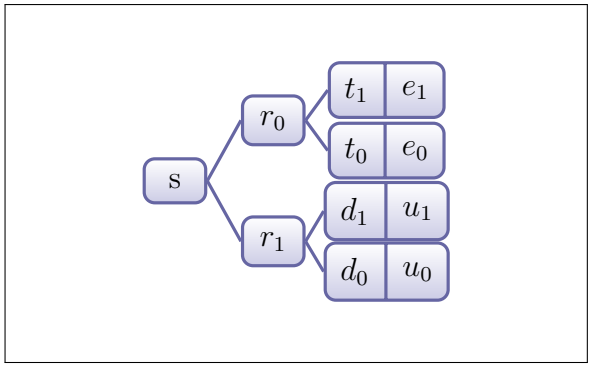

Figura 4.3: *Representação gráfica da nova divisão do segredo em $\mathcal{MQID}$-3A*

Podemos ver então que a equação (4.1) em função de r_1 pode ser reescrita em função de r_0: $\mathcal{G}(r_0, d_1) + u_1 = \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0$. Assim, propomos que o provador demonstre que possui uma tupla $(r_0, r_1, t_0, t_1, e_0, e_1, d_0, d_1, u_0, u_1)$ que satisfaça (4.1 e 4.5) OU (4.4 e 4.5):

$$\mathcal{G}(t_0, r_1) + e_0 = \mathcal{P}(s) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1 \tag{4.1}$$

$$\mathcal{G}(r_0, d_1) + u_1 = \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0 \tag{4.4}$$

$$(t_0, e_0) = (r_0 - t_1, \mathcal{P}(r_0) - e_1) \quad \text{E} \quad (d_0, u_0) = (r_1 - d_1, \mathcal{P}(r_1) - u_1) \tag{4.5}$$

Na figura 4.4 detalhamos o protocolo $\mathcal{MQID}$-3 aprimorado com probabilidade de personificação em uma rodada de 1/2.

Entrada do Provador: $((\mathcal{P}, \mathrm{v}), \mathrm{s})$ Entrada do Verificador: $(\mathcal{P}, \mathrm{v})$

$c_0 \leftarrow Com(r_0, \mathcal{G}(r_0, d_1) + u_1)$
$c_1 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$
$c_2 \leftarrow Com(t_0, e_0)$
$c_3 \leftarrow Com(t_1, e_1)$
$c_4 \leftarrow Com(d_0, u_0)$
$c_5 \leftarrow Com(d_1, u_1)$
$hc \leftarrow H(c_0, c_1, c_2, c_3, c_4, c_5)$

$\xrightarrow{hc}$

Escolhe $Ch \in_R \{0, 1, 2, 3\}$

$\xleftarrow{Ch}$

Se $Ch = 0, Rsp \leftarrow (r_1, t_1, e_1, d_0, u_0, c_0, c_2)$
Se $Ch = 1, Rsp \leftarrow (r_1, t_0, e_0, d_1, u_1, c_0, c_3)$
Se $Ch = 2, Rsp \leftarrow (r_0, t_0, e_0, d_1, u_1, c_1, c_4)$
Se $Ch = 3, Rsp \leftarrow (r_0, t_1, e_1, d_0, u_0, c_1, c_5)$

$\xrightarrow{Rsp}$

Se $Ch = 0$, utiliza $Rsp = (r_1, t_1, e_1, d_0, u_0, c_0, c_2)$ para verificar:
$c'_1 \leftarrow Com(r_1, \mathrm{v} - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1)$
$c'_3 \leftarrow Com(t_1, e_1)$
$c'_4 \leftarrow Com(d_0, u_0)$
$c'_5 \leftarrow Com(r_1 - d_0, \mathcal{P}(r_1) - u_0)$
$hc \stackrel{?}{=} H(c_0, c'_1, c_2, c'_3, c'_4, c'_5)$

Se $Ch = 1$, utiliza $Rsp = (r_1, t_0, e_0, d_1, u_1, c_0, c_3)$ para verificar:
$c'_1 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$
$c'_2 \leftarrow Com(t_0, e_0)$
$c'_4 \leftarrow Com(r_1 - d_1, \mathcal{P}(r_1) - u_1)$
$c'_5 \leftarrow Com(d_1, u_1)$
$hc \stackrel{?}{=} H(c_0, c'_1, c'_2, c_3, c'_4, c'_5)$

Se $Ch = 2$, utiliza $Rsp = (r_0, t_0, e_0, d_1, u_1, c_1, c_4)$ para verificar:
$c'_0 \leftarrow Com(r_0, \mathcal{G}(r_0, d_1) + u_1)$
$c'_2 \leftarrow Com(t_0, e_0)$
$c'_3 \leftarrow Com(r_0 - t_0, \mathcal{P}(r_0) - e_0)$
$c'_5 \leftarrow Com(d_1, u_1)$
$hc \stackrel{?}{=} H(c'_0, c_1, c'_2, c'_3, c_4, c'_5)$

Se $Ch = 3$, utiliza $Rsp = (r_0, t_1, e_1, d_0, u_0, c_1, c_5)$ para verificar:
$c'_0 \leftarrow Com(r_0, \mathrm{v} - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0)$
$c'_2 \leftarrow Com(r_0 - t_1, \mathcal{P}(r_0) - e_1)$
$c'_3 \leftarrow Com(t_1, e_1)$
$c'_4 \leftarrow Com(d_0, u_0)$
$hc \stackrel{?}{=} H(c'_0, c_1, c'_2, c'_3, c'_4, c_5)$

Figura 4.4: *Protocolo de Identificação $\mathcal{MQID}-3$ Aprimorado*

4.2.1 Análise da Proposta

Para a análise do nosso protocolo $\mathcal{MQID}$-3 aprimorado vamos primeiramente explicitar algumas simetrias características de sua estrutura. Os desafios $Ch \in \{0, 1\}$ indicam

que a verificação será realizada em função de r_1, ou seja, utilizando-se as equações 4.1 e 4.5, enquanto temos que se $Ch \in \{2, 3\}$ então a verificação será realizada em função de r_0, ou seja, utilizando-se as equações 4.4 e 4.5. Podemos ainda ressaltar que para as equações 4.1 e 4.4 é efetuada a verificação do lado esquerdo quando $Ch \in \{1, 2\}$ e do lado direito nos demais casos. Este fato estabelece que a chave pública do provador é utilizada durante a verificação apenas se $Ch \in \{0, 3\}$. Na Figura 4.5 vemos um resumo gráfico dessas condições.

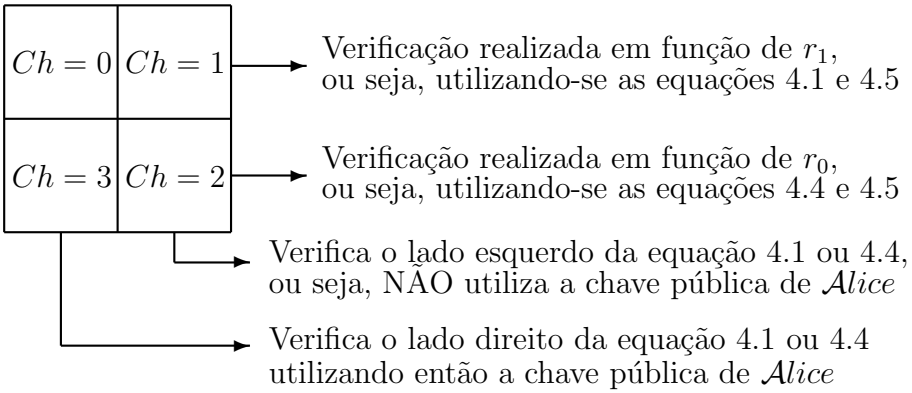

Figura 4.5: *Simetrias características da estrutura de verificação proposta*

Probabilidade 1/2 de Personificação em uma rodada

Percebe-se inicialmente que se um adversário $\mathcal{C}arlos$ executar o protocolo tentando personificar $\mathcal{A}lice$ sem a chave secreta dela, e utilizando um valor s' $\in_R \mathbb{F}^n$ em substituição, ele irá obter sucesso em 1/2 das vezes. Mais precisamente, sempre que $Ch \in \{1, 2\}$, quando o verificador não utiliza a chave pública na checagem e sim os valores enviados pelo próprio provador. É fato que no caso de $\mathcal{C}arlos$ trapacear para conseguir obter sucesso em um dos desafios onde é utilizada a chave pública de $\mathcal{A}lice$, ele inevitavelmente causará a falha em um dos desafios que ele normalmente passaria, ou seja, ele sempre permanecerá com uma chance de personificação igual a 1/2 por rodada. Para ilustrarmos essa situação vamos considerar que $\mathcal{C}arlos$ gere $c_0 \leftarrow Com(r'_0, \mathrm{v} - \mathcal{P}(r'_0) - \mathcal{G}(r'_0, d'_0) - u'_0)$ e $c_1 \leftarrow Com(r'_1, \mathrm{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1)$ utilizando a chave pública v da entidade que ele quer personificar. Neste caso, se $Ch \in \{0, 3\}$ $\mathcal{C}arlos$ terá sucesso na validação, pois o verificador $\mathcal{B}eto$ irá efetuar rigorosamente a mesma conta, obtendo o mesmo resultado. Porém, a chance continua sendo 1/2 visto que não será possível satisfazer, neste caso, as checagens de c_0 e c_1 quando $Ch \in \{1, 2\}$, ocorrendo o mesmo de forma análoga para outras alternativas de trapaça.

Para demonstrarmos o fato que acabamos de exemplificar, qual seja, que um adversário $\mathcal{C}arlos$ tentando forjar a validade dos dados para um determinado desafio acaba, inevitavelmente, por impossibilitar a verificação de um outro, permanecendo sempre com uma probabilidade 1/2 de personificação em uma rodada, vamos resumir as hipóteses de ações

tomadas por $\mathcal{C}arlos$ para tentar alcançar sucesso em verificações que ele normalmente não conseguiria satisfazer e apresentar as consequentes falhas causadas por essas ações que redundam na manutenção constante de uma probabilidade 1/2 de personificação por rodada. Por questão de espaço dividimos esses dados em duas tabelas. Na tabela 4.1 explicitamos as conjecturas de hipóteses de ações tomadas por $\mathcal{C}arlos$ para tentar personificar $\mathcal{A}lice$ perante $\mathcal{B}eto$ e as respectivas consequências dessas ações. Já na tabela 4.2 resumimos os resultados da verificação de cada compromisso para um determinado desafio, considerando em cada linha uma das hipóteses correlacionada pelo mesmo número identificador, sendo os compromissos riscados por um $\times$ aqueles que não conseguem ser validados. Para todos os casos serão utilizados os valores calculados por $\mathcal{C}arlos$ conforme explicado a seguir.

Analisando-se as equações 4.1 e 4.4 vemos que os únicos valores disponíveis para manipulação por um adversário seriam e_0, e_1, u_0 e u_1. Por isso, o adversário $\mathcal{C}arlos$ escolhe s′, como já dissemos para utilizar em substituição a chave secreta de $\mathcal{A}lice$, que ele desconhece, e também escolhe $r'_0, t'_0, d'_0 \in_R \mathbb{F}^n$, $e'_0, u'_0 \in_R \mathbb{F}^m$ e prepara $r'_1 \leftarrow s' - r'_0$, $t'_1 \leftarrow r'_0 - t'_0$, $d'_1 \leftarrow r'_1 - d'_0$, $e'_1 \leftarrow \mathcal{P}(r'_0) - e'_0$ e $u'_1 \leftarrow \mathcal{P}(r'_1) - u'_0$. Como podemos perceber, até este momento $\mathcal{C}arlos$ apenas substituiu a chave secreta de seu alvo por s' e efetuou todos os cálculos conforme previsto no protocolo. Agora, ele irá preparar valores manipulados, utilizando inclusive a chave pública de $\mathcal{A}lice$, para posteriormente tentar vencer qualquer desafio. Assim, ele calcula:

$$\begin{aligned}
\bar{e_1} &\leftarrow v - \mathcal{P}(\text{s'}) + \mathcal{P}(r'_0) - e'_0 \\
\bar{u_1} &\leftarrow v - \mathcal{P}(\text{s'}) + \mathcal{P}(r'_1) - u'_0 \\
\bar{e_0} &\leftarrow v - \mathcal{P}(\text{s'}) + e'_0 \\
\bar{u_0} &\leftarrow v - \mathcal{P}(\text{s'}) + u'_0
\end{aligned}$$

Temos até aqui que os valores $r'_0, r'_1, t'_0, t'_1, d'_0, d'_1, e'_0, e'_1, u'_0$ e u'_1 foram calculados da mesma forma que durante uma execução normal do protocolo. Já $\bar{e_0}, \bar{e_1}, \bar{u_0}$ e $\bar{u_1}$ foram manipulados para que seja possível satisfazer qualquer desafio, como podemos exemplificar aqui: $\mathcal{G}(t'_0, r'_1) + e'_0 \neq \text{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1$, ou seja, chamamos atenção para o fato que utilizando a chave pública de $\mathcal{A}lice$ invalidaria uma verificação como essa, mas $\mathcal{G}(t'_0, r'_1) + e'_0 = \text{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - \bar{e_1}$ seria aceito devido ao valor preparado para $\bar{e_1}$. De modo a não estender em demasia e desnecessariamente essa demonstração, iremos nos ater às situações atinentes às verificações em função de r_1, sendo válido que as situações de verificações em função de r_0 são idênticas, porém ao invés de utilizar um valor $\bar{e_1}$ manipulado seria utilizado um valor $\bar{u_1}$, por exemplo, valendo as demais substituições naturais.

	Hipótese	Consequência
1	*Carlos* executa "normalmente" o protocolo utilizando um valor s' $\in_R \mathbb{F}^n$ em substituição a chave secreta de *Alice* que ele desconhece	Falha se $Ch \in \{0,3\}$, pois o verificador utiliza a chave pública na checagem, sendo $\mathcal{G}(t'_0, r'_1) + e'_0 \neq \mathrm{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1$
2	*Carlos* gera $c_1 \leftarrow Com(r'_1, \mathrm{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1)$	Consegue sucesso se $Ch = 0$, pois o verificador efetua a mesma conta e aceita a verificação, porém passa a não ser possível validar quando $Ch = 1$, mantendo 1/2
3	*Carlos* gera c_1 conforme descrito no item acima e inclui $\bar{e_0}$ em Rsp se $Ch = 1$	O envio de $\bar{e_0}$ em Rsp para $Ch = 1$ faz a verificação de c_1 ser validada, porém gera a reprovação do compromisso c_2, mantendo a reprovação em $Ch = 1$ e a probabilidade 1/2
4	*Carlos* gera c_1 conforme descrito no item 2 acima e utiliza $\bar{e_0}$ ao invés de e'_0	Consegue aprovação nos dois desafios verificados em função de r_1, porém devido a checagem da equação (4.5) os outros dois desafios falham
5	*Carlos* utiliza $\bar{e_1}$ ao invés de e'_1	Novamente consegue aprovação nos dois desafios verificados em função de r_1, mas devido a checagem da equação (4.5) os outros dois desafios falham

Tabela 4.1: *Hipóteses de trapaça para tentar personificar um provador*

Hipótese	$Ch = 0$	$Ch = 1$	$Ch = 2$	$Ch = 3$
1	~~c_1~~, c_3, c_4, c_5	c_1, c_2, c_4, c_5	c_0, c_2, c_3, c_5	~~c_0~~, c_2, c_3, c_4
2	c_1, c_3, c_4, c_5	~~c_1~~, c_2, c_4, c_5	c_0, c_2, c_3, c_5	~~c_0~~, c_2, c_3, c_4
3	c_1, c_3, c_4, c_5	c_1, ~~c_2~~, c_4, c_5	c_0, c_2, c_3, c_5	~~c_0~~, c_2, c_3, c_4
4	c_1, c_3, c_4, c_5	c_1, c_2, c_4, c_5	$c_0, c_2,$ ~~c_3~~, c_5	~~c_0~~, ~~c_2~~, c_3, c_4
5	c_1, c_3, c_4, c_5	c_1, c_2, c_4, c_5	$c_0, c_2,$ ~~c_3~~, c_5	~~c_0~~, ~~c_2~~, c_3, c_4

Tabela 4.2: *Resultados da verificação de cada compromisso para cada desafio, dada uma hipótese de trapaça*

Segurança do esquema de identificação

Passamos agora para a apresentação de três Teoremas concernentes a $\mathcal{MQID}$-3A. Iremos demonstrar que nosso protocolo aprimorado é perfeitamente correto uma vez que uma provadora honesta *Alice*, utilizando sua chave secreta s, sempre será aceita por um verificador honesto *Beto*. Em seguida demonstraremos que nosso protocolo é de conhecimento-zero, visto que um verificador trapaceiro utilizando um simulador $\mathcal{S}$, sem conhecimento da chave secreta, gera uma transcrição do protocolo estatisticamente indistinguível de uma transcrição gerada pela interação de uma provadora honesta *Alice* e um verificador honesto *Beto*. E ainda, que nosso protocolo possui solidez, uma vez que dado duas execuções do protocolo com o mesmo início (ou seja, os mesmos valores de compromisso) e continuações diferentes (ou seja, Ch e Rsp diferentes) é possível descobrir o segredo s ou então o esquema de comprometimento está quebrado, não possuindo vinculação, visto

que teria gerado valores iguais de compromisso a partir de valores diferentes de msg.

Teorema 1 *(Correção) O Protocolo $\mathcal{MQID}$-3 aprimorado possui correção perfeita quando as partes executantes são honestas e respeitam seu projeto.*

Como já visto no Capítulo 2, para que um esquema de identificação seja considerado perfeitamente correto é necessário que todas as execuções realizadas por uma provadora honesta $\mathcal{A}lice$ e um verificador honesto $\mathcal{B}eto$ sempre resultem no aceite da identidade de $\mathcal{A}lice$ por $\mathcal{B}eto$. Nosso teorema estabelece que as partes executantes devem ser honestas, da mesma forma que requisitado pela própria definição do atributo correção. Neste contexto, honesto refere-se ao fato do provador ser efetivamente aquele que diz ser, ou seja, que quem esteja interagindo como provadora no protocolo seja de fato $\mathcal{A}lice$ utilizando sua própria chave privada, conforme previsto pelo esquema. Satisfeito esse requisito evidente, a demonstração da correção de nosso protocolo decorre da própria equação (4.3) para observarmos a validade da verificação da equação (4.1). Já para a equação (4.4), onde utilizamos r_0 para a verificação, temos da mesma maneira que vale o seguinte:

$$\begin{aligned}
\mathcal{G}(r_0, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\
\mathcal{G}(r_0, d_0) + \mathcal{G}(r_0, d_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\
\mathcal{G}(r_0, d_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) - \mathcal{G}(r_0, d_0) \\
\mathcal{G}(r_0, d_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - (u_0 + u_1) - \mathcal{G}(r_0, d_0) \\
\mathcal{G}(r_0, d_1) + u_1 &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0
\end{aligned}$$

As verificações de (4.5) são idênticas ao próprio processo de divisão do segredo pela provadora, nos permitindo concluir assim que $\mathcal{A}lice$ sempre será aceita por $\mathcal{B}eto$ e consequentemente nosso protocolo possui correção perfeita.

Teorema 2 *(Conhecimento-Zero) O Protocolo $\mathcal{MQID}$-3 aprimorado é de conhecimento-zero estatístico quando o esquema de comprometimento $\mathcal{C}om$ possui ocultação estatística.*

Demonstração: Seja $\mathcal{S}$ um simulador que irá personificar uma entidade qualquer utilizando apenas a chave pública v dessa entidade e possuindo acesso a um oráculo $\mathcal{O}$ que indica por meio de um bit se a próxima verificação será em função de r_1 ou de r_0, ou seja, se o próximo desafio a ser recebido será $Ch \in \{0,1\}$ ou $Ch \in \{2,3\}$, respectivamente. A cada rodada do protocolo, $\mathcal{S}$ consulta $\mathcal{O}$ e recebe dele um bit Ch^* indicando que $Ch^* = 1 \rightarrow Ch \in \{0,1\}$, $Ch^* = 0 \rightarrow Ch \in \{2,3\}$. $\mathcal{S}$ então escolhe s', $r'_0, t'_0, d'_0 \in_R \mathbb{F}^n$, $e'_0, u'_0 \in_R \mathbb{F}^m$ e prepara $r'_1 \leftarrow s' - r'_0$, $t'_1 \leftarrow r'_0 - t'_0$ e $d'_1 \leftarrow r'_1 - d'_0$. Se $Ch^* = 0$ então calcula $e'_1 \leftarrow v - \mathcal{P}(s') + \mathcal{P}(r'_0) - e'_0$ e $u'_1 \leftarrow \mathcal{P}(r'_1) - u'_0$, senão $\mathcal{S}$ faz $e'_1 \leftarrow \mathcal{P}(r'_0) - e'_0$ e $u'_1 \leftarrow v - \mathcal{P}(s') + \mathcal{P}(r'_1) - u'_0$. Finalmente, $\mathcal{S}$ gera os valores de compromisso e os envia para o verificador: $c_0 \leftarrow Com(r'_0, \mathcal{G}(r'_0, d'_1) + u'_1)$, $c_1 \leftarrow Com(r'_1, \mathcal{G}(t'_0, r'_1) + e'_0)$, $c_2 \leftarrow Com(t'_0, e'_0)$, $c_3 \leftarrow Com(t'_1, e'_1)$, $c_4 \leftarrow Com(d'_0, u'_0)$ e $c_5 \leftarrow Com(d'_1, u'_1)$.

Após receber Ch, $\mathcal{S}$ responde normalmente com seus valores já calculados, ou seja, se $Ch = 0, Rsp \leftarrow (r'_1, t'_1, e'_1, d'_0, u'_0, c_0, c_2)$, ou se $Ch = 1, Rsp \leftarrow (r'_1, t'_0, e'_0, d'_1, u'_1, c_0, c_3)$, ou se $Ch = 2, Rsp \leftarrow (r'_0, t'_0, e'_0, d'_1, u'_1, c_1, c_4)$, ou se $Ch = 3, Rsp \leftarrow (r'_0, t'_1, e'_1, d'_0, u'_0, c_1, c_5)$.

Verificamos que nos casos em que $\mathcal{O}$ prediz corretamente o desafio, ou seja, em que $Ch^* = 1$ e $Ch \in \{0, 1\}$, ou então, $Ch^* = 0$ e $Ch \in \{2, 3\}$, os valores enviados por $\mathcal{S}$ serão aceitos por qualquer verificador executando normalmente o protocolo. Podemos confirmar o caso concreto quando $Ch^* = 1$ e $Ch = 0$, temos $e'_1 = \mathrm{v} - \mathcal{P}(\mathrm{s}') + \mathcal{P}(r'_0) - e'_0$, logo c_1 ao ser verificado será válido pois teremos então $\mathrm{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1 = \mathcal{G}(t'_0, r'_1) + e_0$. O caso de $Ch^* = 1$ e $Ch = 1$ é mais simples, pois o verificador não utiliza a chave pública v, fazendo a reconstrução dos compromissos com os mesmos valores preparados pelo simulador, o que obviamente redunda no aceite da verificação. Para os casos em que $Ch^* = 0$, e consequentemente $Ch \in \{2, 3\}$, teremos a rigor a mesma situação que acabamos de enumerar, porém com as verificações em função r_0.

Vemos assim que um verificador desonesto utilizando $\mathcal{S}$ pode gerar uma transcrição do protocolo que é indistinguível de uma gerada pela interação de uma provadora honesta *Alice* e um verificador honesto *Beto*, comprovando assim que um adversário não adquire conhecimento útil a partir de transcrições do protocolo, desde que o esquema de comprometimento $\mathcal{C}om$ possua ocultação estatística.

Teorema 3 *(Solidez) O Protocolo $\mathcal{MQID}$-3 aprimorado é argumento de conhecimento com probabilidade de erro 1/2 em uma rodada quando o esquema de comprometimento $\mathcal{C}om$ possui vinculação computacional.*

Demonstração: Sejam $(hc_0, Ch_0, Rsp_0), (hc_1, Ch_1, Rsp_1), (hc_2, Ch_2, Rsp_2)$ e (hc_3, Ch_3, Rsp_3) quatro transcrições da execução de $\mathcal{MQID}$-3 Aprimorado, tais que $Ch_i = i$ e $Dec(\mathcal{P}, \mathrm{v}, hc_i, Ch_i, Rsp_i) = 1$. Sejam $c_0, c_1, c_2, c_3, c_4, c_5$ valores de compromisso gerados de acordo com o estabelecido na definição do protocolo $\mathcal{MQID}$-3 Aprimorado, e ainda, sejam $hc_0 = hc_1 = hc_2 = hc_3 = H(c_0, c_1, c_2, c_3, c_4, c_5)$ e $Rsp_0 = (r_1^{(0)}, t_1^{(0)}, e_1^{(0)}, d_0^{(0)}, u_0^{(0)}, c_0, c_2)$, $Rsp_1 = (r_1^{(1)}, t_0^{(1)}, e_0^{(1)}, d_1^{(1)}, u_1^{(1)}, c_0, c_3)$, $Rsp_2 = (r_0^{(2)}, t_1^{(2)}, e_1^{(2)}, d_0^{(2)}, u_0^{(2)}, c_1, c_4)$ e $Rsp_3 = (r_0^{(3)}, t_0^{(3)}, e_0^{(3)}, d_1^{(3)}, u_1^{(3)}, c_1, c_5)$, temos então que:

$$c_0 = \mathcal{C}om(r_0^{(2)}, \mathcal{G}(r_0^{(2)}, d_1^{(2)}) + u_1^{(2)}) = \mathcal{C}om(r_0^{(3)}, \mathrm{v} - \mathcal{P}(r_0^{(3)}) - \mathcal{G}(r_0^{(3)}, d_0^{(3)}) - u_0^{(3)}) \quad \text{(a)}$$

$$c_1 = \mathcal{C}om(r_1^{(0)}, \mathrm{v} - \mathcal{P}(r_1^{(0)}) - \mathcal{G}(t_1^{(0)}, r_1^{(0)}) - e_1^{(0)}) = \mathcal{C}om(r_1^{(1)}, \mathcal{G}(t_0^{(1)}, r_1^{(1)}) + e_0^{(1)}) \quad \text{(b)}$$

$$c_2 = \mathcal{C}om(t_0^{(1)}, e_0^{(1)}) = \mathcal{C}om(t_0^{(2)}, e_0^{(2)}) = \mathcal{C}om(r_0^{(3)} - t_1^{(3)}, \mathcal{P}(r_0^{(3)}) - e_1^{(3)}) \quad \text{(c)}$$

$$c_3 = \mathcal{C}om(t_1^{(0)}, e_1^{(0)}) = \mathcal{C}om(r_0^{(2)} - t_0^{(2)}, \mathcal{P}(r_0^{(2)}) - e_0^{(2)}) = \mathcal{C}om(t_1^{(3)}, e_1^{(3)}) \quad \text{(d)}$$

$$c_4 = \mathcal{C}om(d_0^{(0)}, u_0^{(0)}) = \mathcal{C}om(r_1^{(1)} - d_1^{(1)}, \mathcal{P}(r_1^{(1)}) - u_1^{(1)}) = \mathcal{C}om(d_0^{(3)}, u_0^{(3)}) \quad \text{(e)}$$

$$c_5 = \mathcal{C}om(r_1^{(0)} - d_0^{(0)}, \mathcal{P}(r_1^{(0)}) - u_0^{(0)}) = \mathcal{C}om(d_1^{(1)}, u_1^{(1)}) = \mathcal{C}om(d_1^{(2)}, u_1^{(2)}) \quad \text{(f)}$$

Das equações (a), (b), (c), (d), (e) e (f) resultam as igualdades a seguir ou então $\mathcal{C}om$ não possui vinculação, visto que teria gerado valores iguais de compromisso a partir de

valores diferentes de msg. Desta forma temos: $r_1^{(0)} = r_1^{(1)}$, $r_0^{(2)} = r_0^{(3)}$, $t_0^{(1)} = t_0^{(2)} = r_0^{(3)} - t_1^{(3)}$, $t_1^{(0)} = r_0^{(2)} - t_0^{(2)} = t_1^{(3)}$, $e_0^{(1)} = e_0^{(2)} = \mathcal{P}(r_0^{(3)}) - e_1^{(3)}$, $e_1^{(0)} = \mathcal{P}(r_0^{(2)}) - e_0^{(2)} = e_1^{(3)}$, $d_0^{(0)} = r_1^{(1)} - d_1^{(1)} = d_0^{(3)}$, $r_1^{(0)} - d_0^{(0)} = d_1^{(1)} = d_1^{(2)}$, $u_0^{(0)} = \mathcal{P}(r_1^{(1)}) - u_1^{(1)} = u_0^{(3)}$, $\mathcal{P}(r_1^{(0)}) - u_0^{(0)} = u_1^{(1)} = u_1^{(2)}$.

A partir de (a) e (b) nós temos que $\mathrm{v} = \mathcal{G}(r_0^{(2)}, d_1^{(2)}) + u_1^{(2)} + \mathcal{P}(r_0^{(3)}) + \mathcal{G}(r_0^{(3)}, d_0^{(3)}) + u_0^{(3)} = \mathcal{G}(t_0^{(1)}, r_1^{(1)}) + e_0^{(1)} + \mathcal{P}(r_1^{(0)}) + \mathcal{G}(t_1^{(0)}, r_1^{(0)}) + e_1^{(0)})$.

Das equações e igualdades citadas acima podemos verificar que:

$$
\begin{aligned}
&t_0^{(1)} + t_1^{(0)} = r_0^{(3)} - t_1^{(3)} + t_1^{(0)} = r_0^{(3)} \\
&e_0^{(1)} + e_1^{(0)} = \mathcal{P}(r_0^{(3)}) - e_1^{(3)} + e_1^{(0)} = \mathcal{P}(r_0^{(3)}) \\
&d_0^{(3)} + d_1^{(2)} = r_1^{(1)} - d_1^{(1)} + d_1^{(2)} = r_1^{(1)} \\
&u_0^{(3)} + u_1^{(2)} = \mathcal{P}(r_1^{(1)}) - u_1^{(1)} + u_1^{(2)} = \mathcal{P}(r_1^{(1)}) \\
&\mathcal{G}(r_0^{(2)}, d_1^{(2)}) + \mathcal{G}(r_0^{(3)}, d_0^{(3)}) = \mathcal{G}(r_0^{(3)}, r_1^{(1)}) \\
&\mathcal{G}(t_0^{(1)}, r_1^{(1)}) + \mathcal{G}(t_1^{(0)}, r_1^{(0)}) = \mathcal{G}(r_0^{(3)}, r_1^{(0)})
\end{aligned}
$$

Chegamos então a $\mathrm{v} = \mathcal{G}(r_0^{(3)}, r_1^{(0)}) + \mathcal{P}(r_0^{(3)}) + \mathcal{P}(r_1^{(0)}) = \mathcal{P}(r_0^{(3)} + r_1^{(0)})$, confirmando assim que poderíamos recuperar a chave secreta s com $r_0^{(3)} + r_1^{(0)}$.

4.2.2 Comparação com o original

Em [55], Sakumoto, Shirai e Hiwatari incluíram uma comparação da eficiência de $\mathcal{MQID}$-3 com outros protocolos de identificação em 3 passos baseados nos problemas SD binário, CLE e PP. Não vimos razão para repetir essa comparação aqui, uma vez que nosso foco, como já dito, é a diminuição da comunicação total do protocolo de Sakumoto *et al*. Por isso, vamos restringir esta análise comparativa ao tamanho da comunicação do nosso protocolo aprimorado ante ao originalmente apresentado.

Em comparação ao protocolo $\mathcal{MQID}$-3 original, vamos demonstrar que nossa versão aprimorada alcança uma comunicação 9,5% menor para os parâmetros sugeridos por Sakumoto *et al*, apesar de aumentar o tráfego em uma rodada. Este ganho é resultado da diminuição do número de rodadas para o mesmo nível de segurança, visto que conseguimos uma probabilidade de personificação em uma rodada de 1/2, contra 2/3 do protocolo $\mathcal{MQID}$-3.

Como já vimos na revisão do protocolo original, o $\mathcal{MQID}$-3 precisa de $\lceil \frac{\lambda}{\lg 3-1} \rceil$ repetições para alcançar uma probabilidade total de personificação menor que $2^{-\lambda}$, porém nosso protocolo aprimorado, por ter uma probabilidade 1/2 em uma rodada, precisa de apenas λ rodadas. Vemos assim que o protocolo original necessita aproximadamente 70,9% mais rodadas para o mesmo nível de segurança.

No protocolo original, lembrando que obrigatoriamente estamos sempre considerando a utilização da extensão de [60], conforme já descrito anteriormente neste texto e também explicitado no último paragrafo do item 3 de [55], temos que o provador gera 3 compromis-

sos e envia um hash hc da concatenação dos mesmos para o verificador, que responde com um desafio Ch do qual decorre o envio de Rsp pelo provador. O hash dos compromissos (hc) possui o tamanho de $2m$ bits, Ch ocupa 2 bits e Rsp é igual a $2n+3m$ bits, totalizando assim $2n + 5m + 2$ bits trafegados em uma rodada. Nosso protocolo gera 6 compromissos inicialmente, efetuando o envio de um hash hc da mesma forma já citada, um desafio de mesmo tamanho e nosso Rsp possui $3n+6m$ bits, perfazendo um total de $3n+8m+2$ bits. Considerando agora a comunicação total para uma execução com probabilidade total de personificação menor que $2^{-\lambda}$, chegamos ao total de $\lambda\cdot(3n+8m+2)$ para o nosso protocolo aprimorado, enquanto o original totaliza $1,709\lambda\cdot(2n+5m+2) = \lambda\cdot(3,42n+8,55m+3,42)$. Considerando n=m temos nossa proposta alcançando um tráfego de dados aproximadamente 8% menor. Utilizando os parâmetros propostos pelos autores originais, $n = 84$ e $m = 80$, $\mathcal{MQID}$-3 dispende 570 bits por rodada, enquanto nosso protocolo utiliza 894, porém, para uma segurança igual a 2^{-30} o protocolo original utiliza 52 rodadas, enquanto o nosso necessita apenas de 30. Assim, teríamos uma comunicação total de 26820 bits com nosso protocolo aprimorado contra 29640 bits de $\mathcal{MQID}$-3, ou seja, uma redução de 9,5%. Incluímos na tabela 4.3 um resumo dos resultados descritos acima com o tamanho da comunicação necessária por cada protocolo.

	Comunicação (bits)	
	$\mathcal{MQID}$-3	$\mathcal{MQID}$-3A
Rodada	$2n + 5m + 2$	$3n + 8m + 2$
Total	$1,71\lambda(2n + 5m + 2)$	$\lambda(3n + 8m + 2)$
	$\lambda(3,42n + 8,55m + 3,42)$	

Tabela 4.3: *Tamanho da comunicação necessária por cada protocolo*

4.2.3 Aprimorando um pouco mais

Lembramos que o parâmetro κ estabelece o nível da segurança da chave secreta e em consequência também os valores de n, m, sendo sempre $m = \kappa$. Além disso, temos no caso do nosso protocolo de identificação um segundo parâmetro λ, utilizado para definir a possibilidade total de personificação, que deverá ser igual ou menor a $2^{-\lambda}$. No caso de um esquema de assinatura derivado de nossa proposta, teríamos então $\lambda = \kappa$. Como já dissemos anteriormente, em virtude da possibilidade de um ataque do aniversário contra o hash hc ou os compromissos gerados, que também são valores de hash, temos que hc e c_i possuem tamanho de 2κ bits cada. Porém, formulamos em co-autoria com o professor Paulo S. L. M. Barreto uma melhoria adicional, que publicamos aqui pela primeira vez e que também pode ser aplicada ao protocolo original, mantendo o mesmo nível de segurança esperado, mas com a utilização de compromissos com tamanhos ligeiramente menores, economizando ainda mais na comunicação total do protocolo. A proposta é manter hc com o tamanho atual de 2κ bits e utilizar compromissos c_i de tamanho igual a $2(\kappa - \lfloor \lg \lambda \rfloor - 2)$

bits, conseguindo, por exemplo, quando temos $\kappa = 80$ e $\lambda = 30$ diminuirmos mais 24 bits por rodada na comunicação (12 bits em cada compromisso incluso em *Rsp*), totalizando uma diminuição de 720 bits no total da comunicação, ou seja, um ganho adicional de aproximadamente 2,6%.

Vamos então detalhar este aprimoramento extra. Um compromisso c_i com comprimento de $2(\kappa - \lg\lambda - 2)$ bits é suscetível a colisão com um esforço de $2^\kappa/4\lambda$, como podemos ver abaixo:

$$\begin{aligned} 2^{\frac{2(\kappa-\lg\lambda-2)}{2}} &= 2^{(\kappa-\lg\lambda-2)} \\ &= \frac{2^\kappa}{2^2\ 2^{\lg\lambda}} \\ &= \frac{2^\kappa}{4\lambda} \end{aligned}$$

Aparentemente não teríamos então o nível de segurança desejado, porém é fato que atacar um único valor de compromisso c_i não possibilita uma verdadeira vantagem ao atacante. Considerando que fosse preciso atacar apenas os 4 compromissos que iriam ser reconstituídos pelo verificador em cada rodada, o atacante necessitaria replicar o ataque do aniversário um total de 4λ vezes, induzindo desta forma uma colisão em *hc*. Outra opção seria atacar diretamente *hc*, que continua com 2κ bits. Em qualquer das duas situações, seria necessário um esforço total de 2^κ bits, como queríamos manter. No caso do nosso $\mathcal{MQID}$-3A, essa melhoria adicional gera uma economia de comunicação igual a $4\lambda \cdot (\lfloor \lg\lambda \rfloor + 2)$, uma vez que trafegamos 2 compromissos reduzidos em cada iteração quando o provador envia *Rsp*.

4.3 $\mathcal{MQID}$-3A com outros níveis de segurança

Como vimos na Seção 3.8, o melhor algoritmo conhecido atualmente para resolver o problema $\mathcal{MQ}$ em $\mathbb{F}_2$ foi apresentado por Bouillaguet *et al.* [6] e necessita de $O(\lg n \cdot 2^{n+2})$ operações binárias, sendo deste modo suficiente um parâmetro de sistema $\mathcal{MQ}(\kappa, \kappa, \mathbb{F}_2)$ para garantir uma segurança de κ bits para a chave privada. Entretanto, Sakumoto, Shirai e Hiwatari preferiram adicionar uma margem de segurança sugerindo um valor de n ligeiramente maior que m, o que nós iremos manter aqui. Para a utilização do protocolo de identificação $\mathcal{MQID}$-3 Aprimorado teríamos ainda um segundo parâmetro de segurança λ para definir uma possibilidade total de personificação igual (ou menor) a $2^{-\lambda}$ resultando em um total de rodadas $r = \lambda$. Para o caso de um esquema de assinatura derivado desse protocolo, precisaríamos ter então r e λ dependentes de κ, ou seja, $r = \lambda = \kappa$. Na Tabela 4.4 apresentamos uma síntese com parâmetros designados para configurações de segurança de 80, 112, 128, 192 e 256 bits, bem como os tamanhos totais do parâmetro do sistema, comunicação quando $\lambda = 30$ e o tamanho de uma assinatura gerada a partir de esquema

criado com o paradigma Fiat-Shamir [28].

κ	n	m	Parâmetro do Sistema	Comunicação $\lambda = 30$	Assinatura ($\lambda = \kappa$)
80	84	80	34,9 KB	26820 bits	7,2 KB
112	120	112	99,3 KB	37740 bits	14,1 KB
128	136	128	145,6 KB	43020 bits	18,4 KB
192	204	192	490,1 KB	64500 bits	41,4 KB
256	282	256	1160,3 KB	85980 bits	73,6 KB

Tabela 4.4: *Parâmetros de $\mathcal{MQID}$-3A de acordo com o parâmetro de segurança κ*

Capítulo 5

Conclusão

Durante o desenvolvimento desta pesquisa procuramos todo o tempo manter nosso foco bem determinado e restrito aos objetivos delineados em nosso planejamento, sempre limitando os assuntos abordados de forma a manter uma concisão que facilita a compreensão do nosso tópico principal. Ao finalizarmos esta dissertação só nos resta enumerar as contribuições efetuadas e apontar possíveis desdobramentos e caminhos futuros a serem explorados.

Após o estabelecimento dos necessários fundamentos e definições básicas, buscamos prover uma rápida e sintética visão dos criptossistemas de chave pública multivariável. Em seguida, ao examinarmos o novo modelo MPKC proposto por Sakumoto, Shirai e Hiwatari é imprescindível destacar sua mais importante característica, que é a segurança da chave secreta dependente exclusivamente do Problema $\mathcal{MQ}$. Este fato possibilita a geração de futuros esquemas de assinatura com segurança demonstrável, o que ainda não havia sido possível com nenhum outro esquema MPKC.

Nossa principal contribuição neste trabalho foi a apresentação de um novo protocolo de identificação canônico, baseado em sistemas polinomiais multivariáveis quadráticos. Nossa proposta foi construída a partir do protocolo de identificação $\mathcal{MQID}$-3, apresentado durante a CRYPTO'2011 por Sakumoto, Shirai e Hiwatari [55], utilizando da mesma forma que o protocolo original a forma polar da função $\mathcal{MQ}$, porém adotando uma divisão diferente da chave secreta que privilegia uma nova simetria na verificação.

Como resultado, alcançamos uma probabilidade de personificação em uma rodada de 1/2, contra os 2/3 originais, elevando o nível de segurança do protocolo e, principalmente, diminuindo o tamanho da comunicação necessária em uma execução. Nossa proposta apresenta uma diminuição de 9,5% no tamanho da comunicação, quando consideramos os parâmetros sugeridos originalmente e também, do mesmo modo que Sakumoto *et al.*, a utilização da técnica sugerida por Stern [60] para reduzir o número de compromissos trafegados, enviando apenas um hash de todos os compromissos no primeiro passo do protocolo e depois incluindo na resposta do provador apenas aqueles compromissos que não podem ser reconstituídos, para que juntamente com os compromissos validados pelo

verificador seja possível efetuar um novo hash para comparar com o recebido inicialmente.

5.1 Trabalhos decorrentes

Como desdobramento das pesquisas realizadas, esta dissertação originou o artigo apresentado junto ao XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'2012) com o título: "Aprimoramento de Protocolo de Identificação Baseado no Problema $\mathcal{MQ}$", disponível nos anais do evento, de autoria de Fábio S. Monteiro, Denise Goya e Routo Terada.

5.2 Pesquisas Futuras

Podemos apontar como consequência imediata, a derivação de um esquema de assinatura digital a partir de nossa proposta, utilizando o paradigma Fiat-Shamir [28]. Na Tabela 4.4 já incluímos o cálculo do tamanho das assinaturas que seriam geradas para cada nível de segurança. Os valores de assinaturas utilizando nossa proposta são ligeiramente maiores que os alcançados com o $\mathcal{MQID}$-3 original, porém há a vantagem de nosso desafio assumir um espaço de saída padrão, não requerendo implementações especiais para o hash utilizado, como acontece no caso da proposta de Sakumoto *et al.* , que por possuir um desafio $Ch \in \{0, 1, 2\}$ precisa adaptar uma função de hash para descartar os valores 3 quando eles ocorrem.

Referências

[1] **Abdalla** *et al.* **(2002)** Michel Abdalla, Jee Hea An, Mihir Bellare e Chanathip Namprempre. From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security. Em *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT'02, páginas 418–433, London, UK. Springer-Verlag. ISBN 3-540-43553-0.

[2] **Alaoui** *et al.* **(2012)** Sidi Mohamed El Yousfi Alaoui, Özgür Dagdelen, Pascal Véron, David Galindo e Pierre-Louis Cayrel. Extended Security Arguments for Signature Schemes, 2012.

[3] **Ars** *et al.* **(2004)** Gwénolé Ars, Jean charles Faugère, Hideki Imai, Mitsuru Kawazoe e Makoto Sugita. Comparison between XL and Gröbner Basis Algorithms. Em *Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security*, AsiaCrypt'2004, páginas 338–353. Springer-Verlag.

[4] **Bellare e Rogaway (1993)** Mihir Bellare e Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. Em *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, páginas 62–73, New York, NY, USA. ACM. ISBN 0-89791-629-8.

[5] **Bernstein** *et al.* **(2009)** Daniel J. Bernstein, Johannes Buchmann e Erik Dahmen, editors. *Post-Quantum Cryptography*. Springer. ISBN 978-3-540-88701-0.

[6] **Bouillaguet** *et al.* **(2010)** Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir e Bo-Yin Yang. Fast Exhaustive Search for Polynomial Systems in $\mathbb{F}_2$. Em Stefan Mangard e François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, páginas 203–218. Springer Berlin / Heidelberg. ISBN 978-3-642-15030-2.

[7] **Bouillaguet** *et al.* **(2011)** Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque e Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. Em *Proceedings of the 14th international conference on Practice and theory in public key cryptography conference on Public key cryptography*, PKC'11, páginas 473–493, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-19378-1.

[8] **Buchberger (1965)** Bruno Buchberger. *An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal*. Tese

de Doutorado, Leopold-Franzens University, Innsbruck. Tradução para inglês realizada por Michael P. Abramson e publicada em 2006 no Journal of Symbolic Computation vol 41.

[9] **Buchmann** *et al.* **(2004)** Johannes Buchmann, Carlos Coronado, Martin Döring, Daniela Engelbert, Christoph Ludwig, Raphael Overbeck, Arthur Schmidt e Ulrich Vollmer. Post-Quantum Signatures. Cryptology ePrint Archive, Report 2004/297, 2004.

[10] **Castro** *et al.* **(2007)** Rafael Dantas Castro, Ricardo Dahab e Augusto Jun Devegili. Introdução à Segurança Demonstrável. Em *Anais do VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, páginas 103–152.

[11] **Chen** *et al.* **(2009)** Anna Inn-Tung Chen, Ming-Shing Chen, Tien-Ren Chen, Chen-Mou Cheng, Jintai Ding, Eric Li-Hsiang Kuo, Frost Yu-Shuang Lee e Bo-Yin Yang. SSE Implementation of Multivariate PKCs on Modern x86 CPUs. Em *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '09, páginas 33–48, Berlin, Heidelberg. Springer-Verlag.

[12] **Courtois** *et al.* **(2000)** Nicolas Courtois, Alexander Klimov, Jacques Patarin e Adi Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. Em *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EuroCrypt'2000, páginas 392–407.

[13] **Courtois (2005)** Nicolas T. Courtois. Short Signatures, Provable Security, Generic Attacks and Computational Security of Multivariate Polynomial Schemes such as HFE, Quartz and Sflash. *IACR Cryptology ePrint Archive*. URL http://eprint.iacr.org/2004/143. Versão extendida e revista do artigo *Generic Attacks and the Security of Quartz* publicado no PKC'2003.

[14] **Courtois** *et al.* **(2001)** Nicolas T. Courtois, Louis Goubin e Jacques Patarin. Quartz, an asymmetric signature scheme for short signatures on PC - Primitive specification and supporting documentation, 2001.

[15] **Courtois** *et al.* **(2003)** Nicolas T. Courtois, Louis Goubin e Jacques Patarin. SFLASHv3, a fast asymmetric signature scheme. *IACR Cryptology ePrint Archive*, 2003:211. URL http://eprint.iacr.org/2003/211.

[16] **Damgård e Nielsen (2008)** Ivan Damgård e Jesper Buus Nielsen. Commitment Schemes and Zero-Knowledge Protocols (2008), 2008.

[17] **Diffie e Hellman (1976)** Whitfield Diffie e Martin E. Hellman. New Directions in Cryptography, 1976.

[18] **Ding e Schmidt (2005)** Jintai Ding e Dieter Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. Em John Ioannidis, Angelos Keromytis e Moti Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, páginas 317–366. Springer Berlin / Heidelberg. ISBN 978-3-540-26223-7.

[19] **Ding** *et al.* **(2006)** Jintai Ding, Jason E. Gower e Dieter Schmidt. *Multivariate Public Key Cryptosystems*, volume 25 of *Advances in Information Security*. Springer. ISBN 978-0-387-32229-2.

[20] **Ding** *et al.* **(2006)** Jintai Ding, Jason E. Gower e Dieter Schmidt. Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over a Finite Field. *IACR Cryptology ePrint Archive*, 2006. URL http://eprint.iacr.org/2006/038.

[21] **Ding** *et al.* **(2007)** Jintai Ding, Christopher Wolf e Bo-Yin Yang. ℓ-Invertible Cycles for Multivariate Quadratic (MQ) Public Key Cryptography. Em *Proceedings of the 10th international conference on Practice and theory in public-key cryptography*, PKC'07, páginas 266–281, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-540-71676-1.

[22] **Dubois** *et al.* **(2007)** Vivien Dubois, Pierre-Alain Fouque, Adi Shamir e Jacques Stern. Practical Cryptanalysis of SFLASH. Em *Proceedings of Advances in Cryptology*, CRYPTO'07. Springer-Verlag.

[23] **Faugère e Joux (2003)** Jean-Charles Faugère e Antoine Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. Em *Proceedings of Advances in Cryptology*, CRYPTO'2003, páginas 44–60. Springer-Verlag.

[24] **Faugère (1999)** Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88. ISSN 0022-4049. URL http://www-salsa.lip6.fr/~jcf/Papers/F99a.pdf.

[25] **Faugère (2002)** Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Em *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, páginas 75–83, New York, NY, USA. ACM. ISBN 1-58113-484-3. URL http://www-salsa.lip6.fr/~jcf/Papers/F02a.pdf.

[26] **Feige** *et al.* **(1988)** Uriel Feige, Amos Fiat e Adi Shamir. Zero-knowledge Proofs of Identity. *Journal of Cryptology*, 1(2):77–94. ISSN 0933-2790.

[27] **Feldmann (2005)** Adam Thomas Feldmann. A Survey of Attacks on Multivariate Cryptosystems. Dissertação de Mestrado, University of Waterloo, Ontario, Canada.

[28] **Fiat e Shamir (1987)** Amos Fiat e Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. Em *Proceedings of Advances in Cryptology–CRYPTO '86*, páginas 186–194, London, UK, UK. Springer-Verlag. ISBN 0-387-18047-8.

[29] **Garey e Johnson (1979)** M. R. Garey e David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. ISBN 0-7167-1044-7.

[30] **Goldreich (2010)** Oded Goldreich. A Short Tutorial of Zero-Knowledge, 2010. (Versão anterior: 2002. Zero-Knowledge twenty years after its invention.).

[31] **Goldwasser** *et al.* **(1985)** Shafi Goldwasser, Silvio Micali e Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). Em *STOC'85*, páginas 291–304.

[32] **Goldwasser** *et al.* **(1988)** Shafi Goldwasser, Silvio Micali e Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Computing*, 17(2):281–308. ISSN 0097-5397. doi: 10.1137/0217017.

[33] **Graham** *et al.* **(1994)** Ronald L. Graham, Donald E. Knuth e Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd ed. ISBN 0201558025.

[34] **Halevi e Micali (1996)** Shai Halevi e Silvio Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. Em *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, páginas 201–215, London, UK, UK. Springer-Verlag. ISBN 3-540-61512-1.

[35] **Hoffstein** *et al.* **(2008)** Jeffrey Hoffstein, Jill Pipher e J.H. Silverman. *An Introduction to Mathematical Cryptography*. Springer Publishing Company, Incorporated, 1 ed. ISBN 0387779930, 9780387779935.

[36] **Katz e Lindell (2008)** J. Katz e Y. Lindell. *Introduction to modern cryptography*. Chapman & Hall/CRC cryptography and network security. Chapman & Hall/CRC. ISBN 9781584885511.

[37] **Kipnis** *et al.* **(1999)** Aviad Kipnis, Jacques Patarin e Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. Em *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EuroCrypt'1999, páginas 206–222. Springer.

[38] **Koblitz (1987)** Neal Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209.

[39] **López e Dahab (2000)** Julio López e Ricardo Dahab. An Overview of Elliptic Curve Cryptography. Relatório técnico, Universidade Estadual de Campinas.

[40] **Mao (2003)** Wenbo Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall Professional Technical Reference. ISBN 0130669431.

[41] **Matsumoto e Imai (1988)** T. Matsumoto e H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. Em *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EuroCrypt'88, páginas 419–453. Springer-Verlag New York, Inc. ISBN 0-387-50251-3.

[42] **Menezes** *et al.* **(1996)** Alfred J. Menezes, Scott A. Vanstone e Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st ed. ISBN 0849385237.

[43] **Mironov (2005)** Ilya Mironov. Hash functions: Theory, attacks, and applications. Relatório Técnico MSR-TR-2005-187, Microsoft Research. URL http://research.microsoft.com/en-us/people/mironov/hash_survey.pdf.

[44] **Nielsen e Chuang (2010)** M.A. Nielsen e I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. ISBN 9781107002173.

[45] **Patarin (2000)** Jacques Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt 88. Em *Des. Codes Cryptography*, volume 20, páginas 175–209, Norwell, MA, USA. Kluwer Academic Publishers. (Versão anterior: 1995. Pag 248–261 em: Don Coppersmith (editor). Advances in Cryptology—CRYPTO 1995, proceedings of the 15th annual international cryptology conference held at the University of California, Santa Barbara, CA, 27–31 Ago 1995. Lecture Notes in Computer Science 963. Springer. ISBN 3-540-60221-6.).

[46] **Patarin (1996)** Jacques Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. Em *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EuroCrypt'96, páginas 33–48.

[47] **Patarin e Goubin (1997)** Jacques Patarin e Louis Goubin. Trapdoor One-Way Permutations and Multivariate Polynomials. Em Yongfei Han, Tatsuaki Okamoto e Sihan Qing, editors, *Information and Communications Security*, volume 1334 of *Lecture Notes in Computer Science*, páginas 356–368. Springer Berlin / Heidelberg. ISBN 978-3-540-63696-0.

[48] **Patarin** *et al.* **(2001)** Jacques Patarin, Nicolas Courtois e Louis Goubin. FLASH, a Fast Multivariate Signature Algorithm. Em *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, CT-RSA 2001, páginas 298–307, London, UK, UK. Springer-Verlag. ISBN 3-540-41898-9.

[49] **Patarin** *et al.* **(2001)** Jacques Patarin, Nicolas T. Courtois e Louis Goubin. QUARTZ, 128-Bit Long Digital Signatures. Em *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, CT-RSA 2001, páginas 282–297. Springer-Verlag.

[50] **Petzoldt** *et al.* **(2011)** Albrecht Petzoldt, Enrico Thomae, Stanislav Bulygin e Christopher Wolf. Small Public Keys and Fast Verification for Multivariate Quadratic Public Key Systems. Em *Proceedings of the 13th international conference on Cryptographic hardware and embedded systems*, CHES'11, páginas 475–490, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-23950-2.

[51] **Pointcheval e Stern (1996)** David Pointcheval e Jacques Stern. Security Proofs for Signature Schemes. Em *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, páginas 387–398, Berlin, Heidelberg. Springer-Verlag. ISBN 3-540-61186-X.

[52] **Pointcheval e Stern (2000)** David Pointcheval e Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptology*, 13(3):361–396.

[53] **Ran Canetti (1998)** Shai Halevi Ran Canetti, Oded Goldreich. The Random Oracle Methodology, Revisited. Cryptology ePrint Archive, Report 1998/011, 1998.

[54] **Rivest** *et al.* **(1978)** Ronald L. Rivest, Adi Shamir e Leonard M. Adleman. A Method for obtaining Digital Signatures and Public-Key Cryptosystems. *CACM*, 21 (2):120–126. (Este é o "RSA paper".).

[55] **Sakumoto** *et al.* **(2011)** Koichi Sakumoto, Taizo Shirai e Harunaga Hiwatari. Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. Em *Advances in Cryptology - CRYPTO'2011*, volume 6841 of *Lecture Notes in Computer Science*, páginas 706–723. Springer Berlin / Heidelberg. ISBN 978-3-642-22791-2.

[56] **Schneier (1995)** Bruce Schneier. *Applied cryptography (2nd ed.): protocols, algorithms, and source code in C*. John Wiley & Sons, Inc. ISBN 0-471-11709-9.

[57] **Shannon (1948)** Claude Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 e 623–656.

[58] **Shor (1997)** Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509.

[59] **Shoup (2008)** Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press. ISBN 9780521516440.

[60] **Stern (2006)** Jacques Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768. ISSN 0018-9448.

[61] **Stern (2011)** Jacques Stern. "Post Quantum"Cryptography: An Imaginary Interview. Slides da apresentação realizada na Advanced School of Cryptography - Campinas, SP, Outubro 2011.

[62] **Terada (2008)** Routo Terada. *Segurança de Dados:Criptografia em rede de computador (2^a ed)*. Edgard Blucher. ISBN 9788521204398.

[63] **Tsujii** *et al.* **(2010)** Shigeo Tsujii, Masahito Gotaishi, Kohtaro Tadaki e Ryou Fujita. Proposal of a Signature Scheme Based on STS Trapdoor. Em *Post-Quantum Cryptography: Proceedings of the Third International Workshop, PQCrypto'2010*, páginas 201–217.

[64] **Wolf (2005)** Christopher Wolf. *Multivariate Quadratic Polynomials in Public Key Cryptography*. Tese de Doutorado, Katholieke Universiteit Leuven.

[65] **Wolf e Preneel (2004)** Christopher Wolf e Bart Preneel. Applications of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Public Key Systems. Cryptology ePrint Archive, Report 2004/263, 2004.

[66] **Wolf e Preneel (2005)** Christopher Wolf e Bart Preneel. Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations. *IACR Cryptology ePrint Archive*, 2005:77.

[67] **Wolf** *et al.* **(2004)** Christopher Wolf, An Braeken e Bart Preneel. Efficient Cryptanalysis of RSE(2)PKC and RSSE(2)PKC. Cryptology ePrint Archive, Report 2004/237, 2004.

[68] **Wolf** *et al.* **(2006)** Christopher Wolf, An Braeken e Bart Preneel. On the security of stepwise triangular systems. *Des. Codes Cryptography*, 40(3):285–302.