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

Yuri da Silva Villas Boas

SRVB Cryptosystem: An attempt to revive Knapsack-based public-key encryption schemes

Florianópolis
2021

Yuri da Silva Villas Boas

SRVB Cryptosystem: An attempt to revive Knapsack-based public-key encryption schemes

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Ciência da Computação.
Supervisor:: Prof. Jean Everson Martina, Dr.

Florianópolis
2021

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

Villas Boas, Yuri
SRVB Cryptosystem : An attempt to revive Knapsack-based public-key encryption schemes / Yuri Villas Boas ; orientador, Jean Everson Martina, 2021.
46 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2021.

Inclui referências.

1. Ciência da Computação. 2. public-Key cryptography. 3. computational complexity. 4. knapsack problem. I. Martina, Jean Everson. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Documento assinado digitalmente
Yuri da Silva Villas Boas
Data: 19/05/2022 19:30:33-0300
CPF: 060.627.039-65
Verifique as assinaturas em https://v.ufsc.br

Yuri da Silva Villas Boas

SRVB Cryptosystem: An attempt to revive Knapsack-based public-key encryption schemes

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

Profa. Thaís Bardini Idalino, Dra.
Departamento de Informática e Estatística da Universidade Federal de Santa Catarina

Prof. Gustavo Souza Banegas, Dr.
Departmento de Ciência e Engenharia da Computação da Universidade de Tecnologia de Chalmers

Prof. Ricardo Felipe Custódio, Dr.
Departamento de Informática e Estatística da Universidade Federal de Santa Catarina

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

Documento assinado digitalmente
PATRICIA DELLA MEA PLENTZ
Data: 20/05/2022 07:34:59-0300
CPF: 959.243.700-97
Verifique as assinaturas em https://v.ufsc.br

Coordenação do Programa de Pós-Graduação

Documento assinado digitalmente
Jean Everson Martina
Data: 20/05/2022 07:20:28-0300
CPF: 003.355.939-25
Verifique as assinaturas em https://v.ufsc.br

Prof. Jean Everson Martina, Dr.
Supervisor:

Florianópolis, 2021.

To my beloved uncle Dr. Sérgio B. Villas-Boas, a truly remarkable man that these brief lines fail to make justice to. To this day, he continues to inspire me to aiming at great goals. His influence helped shape both my career and my attitude towards life's challenges.

"Have a thousand dreams, accomplish a thousand projects, love a thousand people!"
Sérgio Barbosa Villas-Boas (31/03/1964 – 07/05/2016)

ACKNOWLEDGEMENTS

The following people richly deserve individual and grateful mention, for the discriminated reasons:

Maricéia da Silva Villas Boas and Mário Barbosa Villas Boas for having provided me an extraordinary family and exceptional education;

Brazilian tax payers for funding / having funded our tuition and / or salaries;

Staff of the department from maintenance and security, to bureaucracy, to teaching staff, to my advisor, for the incredibly competent, dedicated, patient and accommodating work;

Dr. Charles F. de Barros for having single handed covered the entire session about Shamir's cryptanalysis, and helped with wording and organization of many versions of this text;

Dr. Daniel Santana Rocha for having introduced me to knapsack based public-key cryptography, kindled my interest for the topic, which ultimately lead me to build a career in information security. His enthusiasm and love for pure and applied mathematics and the generosity with which he always shares them with others at any given opportunity is really something to be treasured. Daniel is also the author of one of the two improvements to the original Merkle-Hellman cryptosystem proposed in this work;

Elisa de Oliveira Flemer for contributing with her incredibly solid and precocious experience with editing and for rehearsing and giving insights for the work's public defense.

A very special of thanks is also due to

Onyxcorp® for ceding many working hours from Yuri S. Villas Boas to be dedicated to this work; as well as to

Toptal® for both bringing about this terrific job opportunity and hosting a preliminary version of this work in its blog.

I want to be a Computer Scientist! I want...

to handcraft an algorithm, and industrialize correct results;
to distil a formula, and yield a functionality;
to advocate against myself, and, in failing, deliver my victory;
to bargain with honesty, and cheapen a complexity;
to courageously face the reality of errors, and sanitize a numerical method;
to equip a scientist and help decode a word of Creation;
to conduct a class, and take those boarding it, to an elucidative example, an intuitive analogy, a simple explanation;
to construct a protocol, and allow for the halting of a conflict and the reaching of a consensus;
to help decentralize a power structure, and backup truth and privacy and freedom;

I want to help build, bit by bit, a world in which:
Humans and machines;
Costumers and corporations;
Citizens and states;
all can work together with efficiency, security and in harmony, for individual and common good, and to the
Glory of the Almighty Programmer of the Universe;

I want to be a Computer Scientist!

ABSTRACT

Public-key cryptography is a ubiquitous building-block of modern telecommunication technology. Among the most historically important, the knapsack-based encryption schemes, from the early years of public-key cryptography, performed particularly well in computational resources (time and memory), and mathematical and algorithmic simplicity. Although effective cryptanalyses readily curtailed their widespread adoption to several different attempts, the possibility of actual usage of knapsack-based asymmetric encryption schemes remains unsettled. This Master's dissertation aims to present a novel construction that offers consistent security improvements on knapsack-based cryptography. We propose two improvements upon the original knapsack cryptosystem that address the most important types of attacks: the Diophantine approximations-based attacks and the lattice problems oracle attacks. The proposed defences demonstrably preclude the types of attacks mentioned above, thus contributing to revive knapsack schemes or settle the matter negatively. Finally, we present the Nep.Sec, a contest that is offering a prize for breaking our proposed cryptosystem.

Keywords: Knapsack Problem. Subset Sum Problem. Public-Key Cryptosystem. Merkle-Hellman Cryptosystem. Lattice Oracle. Diophantine Approximation. Shamir Cryptanalysis

RESUMO

Criptografia de chave pública é um elemento onipresente da telecomunicação moderna. Dentre os esquemas de encriptação historicamente importantes, os baseados no 'problema da mochila', já nos primeiros anos da criptografia de chave pública, performaram particularmente bem em termos de recursos computacionais (tempo e memória), e simplicidade matemática e de algoritmo. Embora criptanálises efetivas tenham prontamente impedido a adoção massiva de diferentes tentativas, o problema da possibilidade de uso real de encriptação assimétrica baseada no problema da mochila continua não resolvido. Esta dissertação de Mestrado objetiva apresentar uma nova construção que oferece avanços consistentes em criptografia baseada no problema da mochila. Propomos duas melhorias no criptossistema de mochila original, respondendo aos ataques mais importantes: os baseados nas aproximações Diofantinas e os baseados nos oráculos de problemas de retículos. As defesas propostas demonstradamente evitam tais ataques, o que contribui para ou reviver os esquemas de problema da mochila, ou confirmar a impossibilidade dessa questão. Por fim, apresentamos o Nep.Sec, um concurso oferecendo um prêmio em dinheiro para quem quebrar o criptossistema proposto.

Palavras-chave: Problema da Mochila. Problema da Soma do Subconjunto. Criptossistema de Chave Pública. Criptossistema de Merkle-Hellman. Oráculo de Retículos. Aproximações Diofantinas. Criptanálise de Shamir

RESUMO ESTENDIDO

INTRODUÇÃO

A introdução começa estabelecendo conceitos progressivamente complexos para culminar nas definições criptografia simétrica, esquema de encriptação de chave pública e esquema de assinatura digital. Passamos a listar as definições finais:

Criptografia Simétrica

Uma criptografia simétrica é definida como a 10-upla ordenada

$$\varkappa = (S, d, T, m, P, K, F, A, A', C),$$

onde:

- S é um conjunto de valores de "sementes" (em inglês, "**s**eed") a serem aleatoriamente elicitados, tipicamente grandes inteiros;
- $d: S \to K$ é uma função injetiva para "derivar" (em inglês "**d**erive") elementos de K;
- T é o conjunto de símbolos ou lista ordenada de símbolos de fato empregados na comunicação textual em claro;
- $m: T \to P$ é uma função injetiva (portanto inambígua) o mais natural e trivial possível para "mapear" (em inglês, "**m**ap") fragmentos de textos ("**t**exts") de T para P;
- P é o conjunto de objetos matemáticos referido pelo jargão como texto em claro ou texto plano ("**p**laintext") e que será transformado pelo processo da encriptação;
- K é um conjunto de parâmetros chamados "chaves" ("**k**eys");
- F é uma família de funções ("**f**unctions") paramétricas injetivas (mapeamentos inambíguos) $f_k: P \to C$, com $k \in K$ tipicamente com as propriedades de **confusão** e **difusão**;
- A é uma família de algorítmos de tempo polinomial a_k que tomam $k \in K$ como parâmetros e $p \in P$ como entrada, resultando $f_k(p), \forall (k, p) \in K \times P$;
- $A^\star$ é uma família de algoritmos de tempo polinomial $a_k^\star$ que toma elementos $k \in K$ como parâmetros e $c \in f(P)$ como entradas, resultando em $p \in P$ tal que $f_k(p) = c, \forall (k, c) \in K \times f(P)$;

Encriptação Assimétrica

Um esquema de encriptação de chave assimétrica pode ser definido como uma 12-upla

$$\varkappa^\star = (S, d, T, m, P, K^\star, d^\star, K, F, A, A^\star, C),$$

onde:

1. $S, d, T, m, P, K, F, A, C$ têm as mesmas definições dadas na criptografia simétrica;

2. $d^\star : K^\star \to K$ é uma **bijeção** que pode ser implementada com algoritmo de complexidade polinomial, porém cuja **inversa**, $d^{\star -1} : K \to K^\star$, não;
3. $A^\star$ é uma família de algoritmos paramétricos $a^\star_k$, com complexidade polinomial, que toma elementos $k^\star \in K^\star$ como parâmetros, e $c \in f(P)$ como entradas, e produz $p \in P$ tal que $f_k(p) = c, \forall (k, c) \in K \times f(P)$;
4. Algoritmos paramétricos para computar a pré-imagem p de qualquer entrada $f_k(p) \in f_k(P)$, que não toma como parâmetro fixo um objeto polinomialmente redutíve a k são não-polinomiais. Em outras palavras, saber k (ou algo facilmente redutível a k) é necessário para computar pré-imagens de f_k facilmente.

Esquema de Assinaturas Digitais

Um esquema de assinaturas digitais pode ser definido como uma 13-upla

$$\chi^\star = (S, d, T, m, P, K^\star, d^\star, K, \Sigma, \mathrm{B}, A, A^\star, C),$$

onde:

1. $S, d, T, m, P, K^\star, d^\star, K, C$ têm as mesmas definições dadas no esquema de encriptação assimétrica;
2. Σ é uma família de funções paramétricas $\sigma_{k^\star} : P \to C$ que tomam $k^\star \in K^\star$ como parâmetros. Estas representam as funções de "assinatura" ("**s**igning functions");
3. B é uma família de funções paramétricas $\beta_k : P \times C \to \{0, 1\}$ que tomam elementos $k \in K$ como parâmetros, onde $\beta_{d^\star(k^\star)}(p, c) = 1 \iff c = \sigma_{k^\star}(p)$. Em outras palavras, $\beta_k(p, c)$ especifica se cada c é ou não a assinatura resultante da aplicação de σ tomando como argumento a chave privada $k^\star = d^{\star -1}(k)$ correspondente à pública k, quando aplicada a p. Eles são, portanto, uma função de "verificação" ("**v**erification") de assinaturas;
4. A é uma família de algoritmos paramétricos a_k, de complexidade polinomial, que tomam elementos $k \in K$ como parâmetros e $(p, c) \in P \times C$ como entradas, produzindo $\beta_k(p, c), \forall (k, p, c) \in K \times P \times C$;
5. $A^\star$ é uma família de algoritmos paramétricos $a^\star_k$, de complexidade polinomial, que tomam elementos $k^\star \in K^\star$ como parâmetros e $c \in f(P)$ como entradas, produzindo $p \in P$ tais que $f_k(p) = c, \forall (k, c) \in K \times f(P)$;
6. Algoritmos para computar $sigma_{k^\star}(p)$, que não tomam um parâmetro polinomialmente redutível a $k^\star$ são não polinomiais. Em outras palavras, saber $k^\star$ (ou algo facilmente redutível a $k^\star$) é necessário para computar $\sigma_{k^\star}$ factivelmente.

Motivação, Justificação, Objetivos e Metodologia

Nos demais tópicos da introdução, argumentamos que:

1. O problema da viabilidade de esquemas de encriptação de chave pública ainda está aberto;
2. Muitos criptólogos continuam fazendo tentativas de viabilizar de fato criptossistemas de tal natureza, entre outras razões, porque as primeiras tentativas prometiam enormes vantagens técnias;

3. Criptossistemas de chave pública baseados no problema da mochila são de fácil entendimento, o que:
   a) torna o criptossistema menos obscuro, portanto mais válido do ponto de vista do **princípio de Kerckhoff**;
   b) enseja um excelente tópico de introdução à criptologia, álgebra, complexidade computacional, programação, entre outros, a estudantes;
4. Itens acima justificam o trabalho.

REVISÃO

No capítulo 2, explicamos o criptossistema original de Merkle-Hellman

Parâmetros para o criptossistema Merkle-Hellman original.

dimensionamento (tamanho de bloco)	$N \in \mathbb{N}$
chave privada	$(\mathbf{v}, \alpha, \theta, \pi) \in \mathbb{Z}^N \times \mathbb{Z} \times \mathbb{Z} \times S_N$
chave pública	$\mathbf{u} \in \mathbb{Z}^N$, com $u_i = v_{\pi_i} * \theta \pmod{\alpha}$
plaintext	$\mathbf{b} \in \{0,1\}^N$
ciphertext	$y \in \mathbb{Z}$, com $y = \mathbf{u} \cdot \mathbf{b}$

Decriptação dada por $w = y * \theta^{-1} \% \alpha$ (aqui, % sendo a operação de resto da divisão inteira), seguida da aplicação do *"algoritmo guloso"* (abaixo) e concluída pela permutação $b_i = b'_{\pi^{-1}(i)}$.

Algorithm 1 *Algoritmo guloso* resolvente do problema da soma da subsequência supercrescente **G**reedy (GS4P sigla em inglês)

```
1:  procedure GS4PS(v, w)
2:      w' ← w                                   ▷ inicialização of w'
3:      for i ← N, ··· , 1 do                    ▷ laço sobre os N bits
4:          if w' ≥ v_i then                     ▷ determinação do i-ésimo bit.
5:              b'_i ← 1
6:          else
7:              b'_i ← 0
8:          w' ← w' − v_i * b'_i                 ▷ dedução de v_i do resto
9:      valid_arg ← (w' == 0)                    ▷ verificação da validade de w
10:     return (b', valid_arg)
```

No capítulo 2, também expomos 2 criptanálises ao criptossistema original de Merkle-Hellman. A saber:

Criptanálise de A. Shamir

Envolve aproximações Diofantinas e programação inteira. Está fora do escopo deste resumo.

Criptanálises Baseadas em Oráculos de Problemas de Retículos

Consistem em constatar que a base de retículo L e o vetor incógnito $\mathbf{x}$ abaixo reduzem o problema de se decifrar o texto cifrado ao **problema do menor vetor** em um retículo com base L. O fator δ está ligado à densidade de informação d da instância do criptossistema em questão, a qual precisa ser baixa para que esta criptanálise específica seja aplicável.

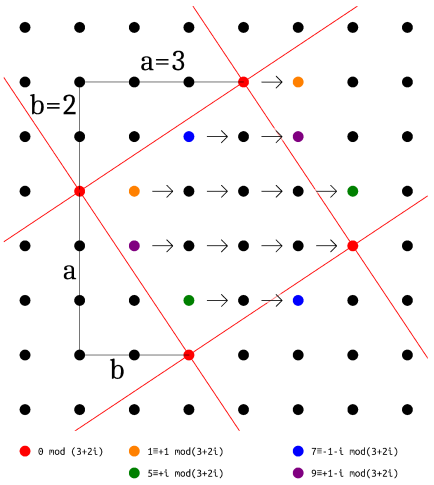
$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ \delta u_1 & \delta u_2 & \cdots & \delta u_n & -\delta y \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ 1 \end{pmatrix} \tag{1}$$

CRIPTOSSISTEMA SRVB

Neste capítulo, apresentamos uma modificação sobre o criptossistema original para cada uma das duas criptanálises mencionadas no capítulo anterior, sendo que também explicamos que ambas podem ser adotadas simultaneamente.

Anel Alternativo

A primeira abordagem, proposta pelo Dr. Daniel Santana Rocha, consiste em adotar inteiros de Gauss como o anel em que são representadas as chaves públicas e os textos cifrados. Tal mudança de anel é possível devido ao isomorfismo exemplificado no diagrama abaixo.



Lattice on the Argand-Gauss plan.

A eficácia desta defesa consiste na impossibilidade de se definir ordem para inteiros de Gauss.

Anel Alternativo

A segunda abordagem, proposta por Yuri da Silva Villas Boas, consiste em generalizar de $N \times 1$ para $n \times N$, com $1 < n < N$ a dimensão das matrizes que representam as cha-

ves, empregando-se, em particular, o corpos finitos isomórficos a $\mathbb{Z}/p$. A eficácea desta defesa consiste na impossibilidade de se definir normas em corpos finitos, sendo que normas são necessárias para a definição dos algoritmos resolventes dos problemas de retículos em que se baseiam as criptanálises.

RESULTADOS OBTIDOS

Os criptossistema proposto consegue:

1. **eliminar ordem** e, portanto, a criptánalise Shamir — Abordagem do Anel Alternativo;
2. **eliminar norma** e, portanto, cripténalises baseadas em oráculos de problemas de retículo — Abordagem de Álgebra Linear sobre Corpos Finitos;
3. **introduzir 'ruído'** ao vetor $\mathbf{u}$ como defesa adicional;
4. **uniformizar (as entradas do) texto cifrado**, com benefícios explicados ao longo do texto;
5. **possibilidade de eliminar evidências estatísticas do valor de** α dada no criptossistema original. Como um bônus, temos também:
6. **análise de complexidade de desempenho real simples**, uma vez que tais resultados são aproximadamente dados por aqueles do criptossistema original multiplicado por n;

Dimensionamento mínimo proposto para o criptossistema é dado por

$$2 \leq n \leq N/2 = 128 = l/3 \tag{2}$$

Potenciais benefícios do prosseguimento deste trabalho, caso criptossistemas baseados no problema da mochila sejam, de fato, viáveis, envolvem o desenvolvimento de um criptossistema com / que:

1. **alta assimetria de complexidade**: $\mathcal{O}(N)$ para encriptação e decriptação, contra, no componente do S4P apenas, $\mathcal{O}(2^{N/2})$ para o melhor ataque clássico determinístico conhecido, $\mathcal{O}(2^{0.291N})$ para o melhor ataque clássico conhecido, e $\mathcal{O}(2^{0.226N})$ para o melhor ataque quântico conhecido. Isso leva a
2. **alta efetividade** do criptossistema se e enquanto criptanálises não são descobertas;
3. **excelente desempenho** em tempo de execução e memória;
4. **exercita o conceito de criptossistema homomórfico** que os criptossistemas baseados no problema da mochila usam como componente; tudo isso, sem deixar de ter
5. **elegante simplicidade**, requerendo conhecimento básico de matemática para ser compreendido, implementado e usado;

Próximos passos envolvem ou requerem a investigação de:

1. **Aprofundamento** da análise de complexidade;
2. **Implementação do SRVB** em linguagens de alto e baixo nível;
3. **Programação do Concurso Nep.Sec** como contrato inteligente em uma blockchain, portanto tornando o Nep.Sec um concurso com requerimento nulo de confiança ‘ganhou-levou’;
4. **Aferição se SRVB é Pós-quântico**;
5. **Adição de um esquema de assinatura digital** baseado no problema vetor mais próximo **CVP** (sigla em inglês), como o GGH ou o NTRUE;
6. **Generalização da Abordagem de Anéis Alternativos** para quatérnios ou mudá-los para inteiros de Eiseinstein;
7. **Generalização da Abordagem de Corpos Finitos** de primos para polinômios com grau maior que 0;
8. **Demonstração** da segurança.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Asymmetric or public-key encryption schemes are highly elaborate constructions both mathematically and conceptually. For the sake of clarity, we shall now start by introducing symmetric encryption schemes first.

1.1 SYMMETRIC ENCRYPTION

An symmetric encryption scheme $\varkappa$ can be defined as the triple of finite, non-empty sets $\varkappa = (K, P, C)$, where $\forall k \in K, k\colon P \to C$ is **injective**, or, in plain English, K is a set of **injective** functions having P as **domain** and C as **codomain**. The idea here is that a user would randomly choose a parameter $k \in K$, appropriately called **k**ey for an algorithm that unambiguously maps elements $p \in P$, called "**p**laintexts", onto $k(p) = c \in C$, called "**c**iphertexts". The rationale for the jargon is exactly what it seems to be: *The 'key' is used both to make* [the meaning of] *a text become 'locked'* (inaccessible or 'ciphered') *and to make* [the meaning of] *a, then, 'locked' text become 'unlocked'* (accessible or 'plain') *again.*

Since k is injective, knowledge of k allows user to calculate the **pre-image** $k^{-1}(c) = p$ of any $c \in k(P)$. A good family of functions K must also bear the properties of **confusion** and **diffusion**, that respectively mean the obfuscation of relation between c and k and c and p — In depth explanation goes beyond the scope of this work. If those are indeed met, knowledge of k are not only sufficient, but also necessary for the attainment of p from c. As a consequence, if the process of selecting k

1. was indeed **true random**;
2. took place secretly; and
3. It is implausible to conceive that any realistic adversary, however well equipped, could possibly determine k by exhaustive trial-and-error of all values in K, typically because $|K|$, number of elements of K, is too large. We will henceforward refer to this property of a set as the it or its elements being **brute-force resistant**;

then user achieves **confidentiality** of p as a result of **secrecy** of k, even if c is made publicly known.

The presented definition could be further nuanced and made more easily understandable by the accretion to this tuple of:

1. T the set of all possible clear **t**exts;
2. $m\colon T \to P$ a **injection** between T and P.

With that, we mean that the elements $t \in T$ consists of the actual (arrays of) *symbols used in a completely clear communication* — like letters, glyphs, arrays of bits, etc. —

while P is a set of *mathematical objects* used by the cryptosystem. m is meant to be the most natural / easily computable possible mapping between the two. Typically, m is publicly known and plays no role in impeding the attainment of t from c. Most notations simply gloss over this definition by just assuming it is trivially implied. For the sake of simplicity of notation, we adopt the same convention.

Similar nuance can also be applied to K, again with benefit of greater clarity. We can define set S of the so called **s**eeds, typically integer true randomly generated numbers, that are then **injectively** (and deterministically) mapped by a **d**eriving function d onto K. This nuance is important because failing to impart injectivity to d, or at least, to make d *as injective as possible — i.e.:* have as few cases of multiple elements $s_1, s_2 \in S$ being mapped to the same $d(s_1) = d(s_2) = k$, and with the lowest multiplicity as possible — means that the resulting keys are actually less random, and therefore more easily guessable by trial-and-error, or, in other words, less secret, than possible.

Another aspect about instantiation of keys that worth mentioning is that it is critical that the algorithm for computing $d(\cdot)$ (as well as that for drawing s, but this part tends to be trivially true) has to be as close to uniform in time, memory and energy resources as possible in order to avoid **side-channel attacks**. The same applies for any other cryptographic algorithm or any security critical algorithm, for the same reasons. Although worth mentioning and indeed critical from the point of view of *implementation and real use*, it is important to clarify that side-channel attacks and defenses for them fall in the intersection of information security and computer engineering, and therefore in a scope posterior to that of initial mathematical cryptologic conceptualization.

An attentive, more familiarized reader could, at this point, object that our proposed definition fails to encompass and present the concepts of **initialization vector**, or **message authenticating code**. While that is true, both concepts are **not** strictly necessary for the definition of a symmetric encryption scheme. Furthermore, messages imbued with one or both of these attributes **can** be encompassed as belonging to subsets of P bearing specific properties. Therefore, the concepts of initialization vector and message authentication code are left outside of the scope of this work.

Finally, a more critical objection is that nothing, so far, has been said about computation of k or k^{-1}, and the notions of *parameters* for a function or algorithm, *parameterized function* and *parameterized algorithm* have been used interchangeably. This glossing over can be safely done in the context of symmetric encryption, for which knowledge of the key k employed for producing ciphertext $c = k(p)$ correspondent to plaintext p is the necessary, sufficient condition for retrieving the latter. As it will be clear soon, however, the nuances between these concepts lie at the very core of asymmetric cryptography. The only remaining caveat to be made is that, likewise with the implementation of the algorithms for computing $d(\cdot)$, those for $k(\cdot)$ and $k^{-1}(\cdot)$ are supposed to address concerns about side-channel attacks.

To summarize, an in detail definition of symmetric encryption scheme would be a tuple

$$\varkappa = (S, d, T, m, P, K, F, A, A', C),$$

where:

- S is a set of "**s**eed" values to be true randomly drawn, typically large integer numbers;
- $d : S \to K$ is an injective function for "**d**eriving" elements of K;
- T is the set of symbols or arrays of symbols employed in an actually clear **t**extual communication;
- $m : T \to P$ is a trivial, as natural as possible injective (therefore unambiguous) **m**apping between the (fragments of) **t**exts of T and elements of P;
- P is the set of mathematical objects, referred to as "**p**laintexts" to be actually transformed by encryption;
- K is a set of parameters called **k**eys;
- F is a family of parametric injective **f**unctions (*i.e.:* unambiguous mappings) $f_k : P \to C$, with $k \in K$ typically bearing the properties of **confusion** and **diffusion**;
- A is a family of polynomial time parametric algorithms a_k, taking elements $k \in K$ as parameters, and $p \in P$ as inputs and yielding $f_k(p), \forall (k, p) \in K \times P$;
- $A^\star$ is a family of polynomial time parametric algorithms $a_k^\star$, taking elements $k \in K$ as parameters, and $c \in f(P)$ as inputs and yielding $p \in P$ such that $f_k(p) = c, \forall (k, c) \in K \times f(P)$;

1.2 ASYMMETRIC ENCRYPTION

A condensed definition of asymmetric encryption scheme, in the same lines of that given to symmetric encryption, given above, would be a tuple

$$\varkappa^\star = (S, d, T, m, P, K^\star, d^\star, K, F, A, A^\star, C),$$

where:

1. $S, d, T, m, P, K, F, A, C$ have the same definitions as in symmetric cryptography;
2. $d^\star : K^\star \to K$ is a **bijection** that can be implemented with an algorithm of polynomial complexity (*i.e.:* can be easily computed), but whose **inverse**, $d^{\star -1} : K \to K^\star$, can't;

3. $A^\star$ is a family of parametric algorithms $a^\star_k$, of polynomial complexity, taking elements $k^\star \in K^\star$ as parameters, and $c \in f(P)$ as inputs and yielding $p \in P$ such that $f_k(p) = c, \forall (k, c) \in K \times f(P)$;

4. Parametric algorithms for computing the pre-image p of any input $f_k(p) \in f_k(P)$, not taking as input a parameter polynomially reducible to k are non-polynomial. In other words, knowing k (or anything easily reducible to k) is necessary to compute pre-images of f_k easily.

1.3 DIGITAL SIGNATURE SCHEMES

Even though our work is not about signature schemes, for the sake of completion, we will provide a likewise condensed definition of digital signature scheme, given by a tuple

$$\chi^\star = (S, d, T, m, P, K^\star, d^\star, K, \Sigma, \mathrm{B}, A, A^\star, C),$$

where:

1. $S, d, T, m, P, K^\star, d^\star, K, C$ have the same definitions as in asymmetric encryption schemes;

2. Σ is a family of parametric functions $\sigma_{k^\star} : P \to C$ taking elements $k^\star \in K^\star$ as parameters. These are meant to be the **s**igning function;

3. B is a family of parametric functions $\beta_k : P \times C \to \{0, 1\}$ taking elements $k \in K$ as parameters, where $\beta_{d^\star(k^\star)}(p, c) = 1 \iff c = \sigma_{k^\star}(p)$. In other words, $\beta_k(p, c)$ tells whether c is the signature yielded by the signing function σ taking as argument the private key $k^\star = d^{\star -1}(k)$ correspondent to the public k, when applied to p. They are, therefore, meant to **v**erify signatures;

4. A is a family of parametric algorithms a_k, of polynomial complexity, taking elements $k \in K$ as parameters, and $(p, c) \in P \times C$ as inputs and yielding $\beta_k(p, c)$, $\forall (k, p, c) \in K \times P \times C$;

5. $A^\star$ is a family of parametric algorithms $a^\star_k$, of polynomial complexity, taking elements $k^\star \in K^\star$ as parameters, and $c \in f(P)$ as inputs and yielding $p \in P$ such that $f_k(p) = c, \forall (k, c) \in K \times f(P)$;

6. Parametric algorithms for computing $sigma_{k^\star}(p)$, not taking as input a parameter polynomially reducible to $k^\star$ are non-polynomial. In other words, knowing $k^\star$ (or anything easily reducible to $k^\star$) is necessary to feasibly compute $\sigma_{k^\star}$.

1.4 MOTIVATION

Although more than 40 years have passed since the (public) introduction of public-key cryptosystems by Whitfield Diffie and Martin Hellman (DIFFIE; HELLMAN, 1976) (in their case, a *key agreement scheme*), they continue to be an arcane field of knowledge for most people, and yet their importance in contemporary world can hardly be overstated. From a simple internet browsing to all forms of online or digital payment, to digital documents, to cryptocurrencies and online voting systems, they are all made possible by the nearly ubiquitous — and, nonetheless, grossly underused — powers of public-key cryptosystems.

Due to their inherent complexity, public-key cryptosystems usually represent the critical (non-human) component in the systems they are part of, regarding development, implementation and run-time costs and effectiveness. Public-key cryptography is, therefore, a field of both relevance to society as well as a challenging and intellectually demanding field of knowledge. It provides immediate practical applications for some of the most abstract fields in mathematics and theory of computation, stretches through software and electronic engineering, and causes direct, significant impact in delicate and ever evolving balances between individuals and collectivities.

1.5 JUSTIFICATION

In 1978, two years after the foundational work by Diffie and Hellman (DIFFIE; HELLMAN, 1976), the latter and Ralph Merkle devised a public-key cryptosystem based on a particular case of the *Knapsack Problem*, the *Subset Sum Problem*, SSP for short (MERKLE; HELLMAN, 1978). The cryptosystem became widely known as the Merkle-Hellman, or simply MH Cryptosystem.

Actual implementations of the Merkle-Hellman cryptosystem proved it to have an excellent time and memory performance (ODLYZKO, 1990). Many cryptologists were doubtful about its reliability from the beginning (ODLYZKO, 1990), and four years after its creation, the MH cryptosystem was broken with an attack by A. Shamir(SHAMIR, 1982). There were several more attempts to revive MH, many of which also failed for either not qualitatively changing the possible lines of attack or proving to be vulnerable to other types of attacks. Nevertheless, whether or not the usage of a knapsack-based asymmetric cryptosystem has any future is unsettled. Due to its inherent qualities, new attempts continue to appear to revive the concept (WANG; HU, 2010) (THANGAVEL; VARALAKSHMI, 2017).

Progress in either direction will therefore be of keen relevance to the field of cryptology: If knapsack-based cryptosystems prove to be viable, a highly competitive algorithm will become available to *ciphersuites* in web browsers all over the internet. If, conversely, the viability of knapsack-based cryptosystems is at last disproved, valuable

scientific resources will be made available for other areas of computer science.

Another benefit, however indirect, can be claimed to derive from the development and vulgarization of attempts for public-key cryptosystem based on *mathematically simple public-key cryptosystems*, as are the knapsack-based ones: mass adoption of complex information technologies that have public-key cryptosystems at their core will be helped by the public being made more familiar to public-key cryptography in the first place.

That in turn could allow entire infrastructures that, without a cyber-security educated public will remain impossible. Consider, for example, that if a vast majority of internet users understood the concept of digital signature, the need for passwords for accounts of multiple web services could be advantageously replaced by the management of much fewer private keys.

1.6 OBJECTIVES

Our main goal is to help assess the long sought-after viability of knapsack-based public-key cryptosystems. We present two original improvements upon the original knapsack cryptosystem that address the most important types of attacks: the Diophantine approximations-based attacks and the lattice problems-oracle attacks.

To that point, it is unclear which direction, viability or nonviability of knapsack-based cryptosystems each proposed improvements points to. As explained above, either outcome would be beneficial for the field and, therefore, our work does seem to contribute, however anecdotally, to the elucidation of this question.

As a secondary objective, we also aim at producing a high school–accessible content on public-key cryptography that could, therefore, contribute to the popularization of this field.

1.7 METHODOLOGY

Experience shows that it is exceedingly difficult to prove the **in**existence of low complexity cryptanalyses to one given public-key cyrptosystem. A typical public-key cryptosystem's life cycle, therefore, is shaped by constant *educated guessing* by the community of experts at large on the likelihood of cryptanalyses first: existing; and second: given that they exist, being discovered, soon enough to curtail the actual employment of the cryptosystem. As we shall see, this helps establish an alignment of incentives that favours availability and preservation of technological and scientific knowledge.

As knowledge applicable to the cryptanalysis of each given cryptosystem is accumulated over time, deciphering methods with increasingly low complexity arise and the perceived likelihoods of a polynomial cryptanalysis existing and being found

soon rises. Another way to describe the life cycle of a public-key cryptosytem is that it can only be deemed secure if and when there is enough knowledge to conclude effective cryptanalyses for it are (if possible) very difficult to be found, but not enough to suspect such cryptanalysis is remotely likely to be found soon. Finally, in order for a cryptosystem to be deemed *viable*, it must also be competitive regarding the complexity of the best deciphering methods when compared to other cryptosystems, as well as the consumption of computational resources.

Each cryptologist being both A: prompted to advance and, particularly, by Kerckhoff's principle, publish knowledge on whatever given cryptanalysis; and B: unable to prevent any other cryptologist from doing the same (aiming, for instance, at extending the window of secure usage of the given cryptosystem); causes a Nash equilibrium in which the individuals (attempt to) do exactly that. This guarantees that the community as a whole will never cease attempting to close the window of secure employment of any cryptosystem. Conversely, the latter fact magnifies the importance of constant development of new cryptographic techniques. Therefore the field of cryptology, due to the nature of its objects and incentives to agents involved in creating, analysing, and using them is particularly favorable to disclosure and preservation, testing and updating of acquired knowledge.

1.8 PUBLICATIONS

BOAS, Yuri da Silva Villas et al. **F2MH Cryptosystem: Preliminary analysis of an original attempt to revive Knapsack-based public-key encryption schemes.** In: 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2020. p. 211-215. - Qualis A2

BOAS, Yuri da Silva Villas et al. **SRVB cryptosystem: another attempt to revive Knapsack-based public-key encryption schemes** In:27th International Conference on Telecommunications (ICT) on 5-7 October, 2020. - Qualis B1

1.9 STRUCTURE OF THE DOCUMENT

We organize the dissertation as follows:

Chapter 2: revises the original Merkle-Hellman cryptosystem (3.1) and its original cryptanalysis (3.2).

Chapter 3: presents the two original proposals for defending MH cryptosystem against cryptanalyses based on diophantine approximations and lattice-problem oracle.

Chapter 4: presents a quantitative analysis of security improvements of said proposals compared to the original MH.

Chapter 5: conclusion and proposed next steps.

2 MERKLE-HELLMAN KNAPSACK CRYPTOSYSTEM AND CRYPTANALYSES

MH cryptosystem is a very mathematically elegant and simple public-key cryptosystem, indeed requiring only high school level algebra to be understood. In a few sentences, it exploits the difficulty of the **S**ub**S**et **S**um **P**roblem (S3P). S3P can be phrased as follows: *Find out which subset* —here, the plaintext—*, from a certain set of parcels* —the public key—*, has its elements totaling a given value* —the ciphertext. As we shall see in session 4, without the addition of further properties, the best known solutions for this problem have exponential time complexity even for quantum computation.

Even though the classic definition of the problem refers to subsets of parcels, actual implementations of cryptosystems involving it can more easily and clearly be specified with scalar multiplication of a *vector* of parcels with one of zeroes and ones. Those two definitions will, henceforward be used interchangeably.

The following is one instance of S3P: *Determine what subset* $X \subseteq S$ *has parcels summing up* y*, for:* $S = \{896, 48, 964, 89, 648, 96, 489, 64\}$*, and* $y = 2461$*.*

A very strong property is required for another set of parcels, that is part of private key's. This property grants both the existence of a **linear time** algorithm for solving S3P and unicity of the solution (*ie.:* that the ciphertext is unambiguous), and it can be referred to as "superincreasingness" or the parcels from a sequence being "superincreasing". We will henceforward refer to **S**superincreasing S3P's as S4PThe other components of the private key are:

1. a brute-force resistant parameters θ, α for a function $\mathscr{F}$ that maps the private parcels into the public ones destroying superincreasingness while preserving sums (*i.e.:* $\forall v_1, v_2, \mathscr{F}(v_1 + v_2) = \mathscr{F}(v_1) + \mathscr{F}(v_2)$); and

2. a brute-force resistant permutation $\pi \in S_N$ of the order of the public parcels in the public vector relation to their pre-images in the private one. That contributes to further obfuscate the instance of said function.

In other words, the sender, say, Bob draws parcels from the *hard set* — a set devoid of superincreasingness — and proceeds to send their sum to the key's owner, say Alice. Alice uses her private key to map the received sum onto the sum that would be obtained, had Bob used the parcels of the *easy set* instead — the set having superincreasing parcels. Alice, then, exploits that property to ascertain which *easy parcels* those were, and consequently which one subset of corresponding hard parcels were chosen by Bob. Ergo, Bob is able to communicate with Alice by encoding his plaintext onto subsets of *hard parcels*, whilst an eavesdropper, say, Eve, has no feasible means to decipher Bob's ciphertexts.

We shall, now, look into MH and the two main cryptanalyses:

2.1 THE MERKLE-HELLMAN KNAPSACK CRYPTOSYSTEM

For the basic Merkle-Hellman cryptosystem, the following parameters must be chosen:

Table 1 – Parameters for the original Merkle-Hellman Cryptosystem.

sizing (block size)	$N \in \mathbb{N}$
private key	$(\mathbf{v}, \alpha, \theta, \pi) \in \mathbb{Z}^N \times \mathbb{Z} \times \mathbb{Z} \times S_N$
public key	$\mathbf{u} \in \mathbb{Z}^N$
plaintext	$\mathbf{b} \in \{0,1\}^N$
ciphertext	$y \in \mathbb{Z}$

Table 1 contains the parameters for original MH cryptosystem. Here, S_N denotes the symmetric group over $\{1, \cdots, N\}$. In other words, π is a randomly selected permutation of N elements (one of the $N!$ possible). $\mathbf{v}$ is the private, superincreasing vector. Superincreasingness of $\mathbf{v}$ means:

$$\forall i, 0 < \sum_{j=1}^{i-1} v_j < v_i. \tag{3}$$

Namely, the value of each component is positive and strictly greater than the sum of all previous ones. Parameter α must satisfy

$$\sum_{i=1}^{N} v_i < \alpha, \tag{4}$$

and

$$gcd(\alpha, \theta) = 1. \tag{5}$$

$\mathbf{u}$ is the public vector, given by:

$$u_i = \mathscr{F}_\theta(v_{\pi(i)}) \tag{6}$$

with

$$\mathscr{F}_\theta(v) = v * \theta \pmod{\alpha} \tag{7}$$

In summary, we want a *homomorphic encryption* — an encryption that preserves an operation — function $\mathscr{F}_\theta : \mathbb{Z}/\alpha \to \mathbb{Z}/\alpha$, that is a *permutation* of $\mathbb{Z}/\alpha$ aiming at obfuscating the *superincreasingness* of the sequence of (the natural integers corresponding to each of) the v_i's. The reason for (4) is ensuring that all possible encrypted messages do 'fit' in the codomain $\mathbb{Z}/\alpha$ of encrypting function (*ie.*: encryption is injective), whereas (5) is we will need a θ^{-1} that ensures there exists a $\mathscr{F}_\theta^{-1} = \mathscr{F}_{\theta^{-1}}$, which will be used in the decryption.

Optimal key distribution should aim at

$$\forall i,\, v_i \approx 2^{N+i} \text{ and } \alpha \approx 2^{2*N+1}. \tag{8}$$

The rationale is we have to accommodate not having an unnecessarily 'sparse' super-increasing sequence with not having a too short v_1, which would, otherwise, be easy to guess by blind trial and error (ODLYZKO, 1990). We also must have θ uniformly distributed among $\{x \in \mathbb{Z}/\alpha | gcd(x, \alpha) = 1\}$.

Encryption of plaintext $\mathbf{b}$ is given by:

$$y = \mathbf{u} \cdot \mathbf{b} \in \mathbb{Z} = \sum_{i=1}^{N} u_i * b_i. \tag{9}$$

In other words, user performs a scalar multiplication between public key and plain text.

Decryption of ciphertext $\mathbf{y}$ is given by the following 3 steps:

1. First, we **compute** $w = \mathscr{F}_\theta^{-1}(y)$ given by

$$\begin{aligned} w = \mathscr{F}_\theta^{-1}(y) &\overset{(7)}{=} y * \theta^{-1} \overset{(9)}{=} (\sum_{j=1}^{N} u_j * b_j) * \theta^{-1} = \\ \sum_{j=1}^{N} (u_j * b_j * \theta^{-1}) &\overset{(6)}{=} \sum_{j=1}^{N} (v_{\pi(j)} * \theta * \theta^{-1} * b_j) = \\ \sum_{j=1}^{N} (v_{\pi(j)} * b_j) &\overset{j=\pi^{-1}(i)}{=} \sum_{\pi^{-1}(i)=1}^{N} (v_{\pi\circ\pi^{-1}(i)} * b_{\pi^{-1}(i)}) = \\ &\sum_{\pi^{-1}(i)=1}^{N} (v_i * b_{\pi^{-1}(i)}) \pmod{\alpha} \end{aligned} \tag{10}$$

   In other words, by computing $w = \mathscr{F}_\theta^{-1}(y)$ we find exactly the subset sum that the sender would have obtained had they operated in the private vector $\mathbf{v}$, with secret permutation π of correspondent entries also being taken into consideration.

2. Next, we must determine that correspondent entries selection and store it in a variable, say, $\mathbf{b}'$. In other words, we **solve S4P**. That is done by applying a greedy algorithm (described below at Algorithm 2):

Algorithm 2 Greedy **S**ub **S**uperincreasing **S**equence **S**um **P**roblem (S4P) **S**olver

```
1: procedure GS4PS(v, w)
2:     w' ← w                                   ▷ initialization of w'
3:     for i ← N, ··· , 1 do                    ▷ descending loop on the N bits
4:         if w' ≥ v_i then                     ▷ Determine the i-th bit.
5:             b'_i ← 1
6:         else
7:             b'_i ← 0
8:         w' ← w' − v_i ∗ b'_i                 ▷ deduct v_i from the remainder
9:     valid_arg ← (w' == 0)                    ▷ Verify validity of w
10:    return (b', valid_arg)
```

Algorithm 2, above, describes a linear time solution for S4P and takes as input the recently obtained w and the private superincreasing vector $\mathbf{v}$. The obtained output to a valid input (an actual plaintext produced with public key) has:

$$b'_i = b_{\pi^{-1}(i)} \tag{11}$$

3. Finally, we proceed as in the last two lines of (10) to find

$$b_i = b'_{\pi(i)}, \tag{12}$$

or, in other words, undo the effect of permutation to ciphertext, again with linear time.

Now, the explanation of why Algorithm 2 works: (GS4PS), is a, so called **greedy algorithm**, which is a loose jargon to refer to algorithms that, in some sense have the heuristic of 'doing the most possible at each step or iteration'. Here, we iterate exactly once on each parcel v_i of the private vector $\mathbf{v}$, *from the highest* (v_N) *to the lowest* (v_1), and at each such step, we ascertain whether or not that given parcel had been used in the sum that yielded the ciphertext — hence the classification.

We start by noting that we have a remainder variable w', that we initially set to be w (line 2), which, from (10), equals what the sender would have obtain had they operated with the parcels of the private key that correspond exactly with those they actually used. The criterion adopted at every step is straightforward: *if and only if* the parcel v_i currently being considered can be used to sum up w' (line 4), then assume that was the case, that is, *i.e.:* assign $b_i = 1$ and discount v_i from w, and do the opposite — $b_i = 0$ and do **not** discount v_i from w — otherwise (or, in other to achieve uniformity of number of operations independently of input, deduct $v_i * b_i$, anyway as indicated by line 8).

By performing the deduction of w' referred to in the previous paragraph, receiver:

1. extends the validity of premise of the criterion used in steps with $i \in \{N, \cdots, 2\}$;

2. provides for him or herself a criterion for determining the validity of the received ciphertext. Namely, after iteration of $i = 1$ is done, the referred premise implies that all contributing parcels of w were discounted exactly once, and therefore w' must be zero. This verification is done by assigning to variable `valid_arg` the truth value of $w' = 0$ at line 9.

At last, we shall now demonstrate the correctness of the premise that allows for criterion of line 4. From $b'_j \in \{0, 1\}$ and the *superincreasingness*, if $v_i \leq w'$, we have:

$$\forall\, 1 \leq i \leq N, \sum_{j=1}^{i-1} b'_j * v_j \leq \sum_{j=1}^{i-1} v_j \overset{(3)}{<} v_i \leq w' \tag{13}$$

and, therefore, $\sum_{j=1}^{i-1} b'_j * v_j < w'$. In other words, we know that any subset of $\{v_1, \cdots, v_i\}$ **lacking** v_i will sum up strictly less than w', thus **if** there is a solution, it must **include** v_i (*i.e.:* $b'_i = 1$). Conversely, if $v_i > w'$, since all parcels are non-negative (and, in particular, positive), any subset including v_i sum up strictly (even) more than w', and therefore **if** there is solution, it must **exclude** v_i (*i.e.:* $b'_i = 0$).

In other words, (13) allows us to, at each step, disregard parcels smaller than the one currently under consideration as *too small to matter*. Parcels are considered in descending order, and therefore that applies for all parcels yet to be considered. Previously considered parcels no longer matter either because their possible effect has already been factored in by the deduction of w' (in line 8), and, in fact, that deduction makes sure that, at any given point of the iteration of the loop for a valid input $w \in \mathbf{v} \cdot \{0.1\}^N$, w' is strictly less than any previously considered parcel, which confirm that those parcels no longer can contribute to sum up w'. Every factor is, therefore, irrelevant to the determination of every other. After considering each of them individually exactly once, we can conclude that there is no correct solution having any b_i different from what was determined on its corresponding step. Finally, by verifying whether or not $w' = w - \mathbf{v} \cdot \mathbf{b} = 0$ at that point, we can also know if this only possible solution for $\mathbf{b}$ is indeed a solution at all.

Ergo, any $\mathbf{b} \in \{0, 1\}^N$ yields an unambiguous w, and the algorithm is sure to correctly decrypt it. Conversely, if an invalid w (one such $w \notin \mathbf{v} \cdot \{0, 1\}^N$) is provided, the algorithm will detect its invalidity.

Another way to prove the injectivity of T is as follows: Consider the comparison of the summations

$$w_1 = \sum_{i \in s_1} v_i \ , \quad w_2 = \sum_{i \in s_2} v_i$$

of the entries of $\mathbf{v}$ indexed by the elements of two subsets $s_1, s_2 \subseteq \{1, \cdots, N\}$. Let all common parcels (those of indexes belonging to $s_1 \bigcap s_2$) be cancelled yielding an

equivalent comparison between the summations

$$w'_1 = \sum_{i \in s'_1} v_i \ , \quad w'_2 = \sum_{i \in s'_2} v_i$$

of the elements of the correspondent subsets $s'_1 = s_1 - s_2$ and $s'_2 = s_2 - s_1$ of remaining entries. If $w'_1 = w'_2 = 0$, trivially $s_1 \subseteq s_2 \subseteq s_1 \implies s_1 = s_2$ and there is nothing to argue. In the opposite case, let us call $M \in \{1, 2\}$ the set index of that s'_x containing $i_{max} = \max(s'_1 \cup s'_2)$, and m the other. From the *superincreasingness*, we know that

$$w'_m = \sum_{j \in s_m} v_j \leq \sum_{j < i_{max}} v_j \overset{(3)}{<} v_{i_{max}} \leq \sum_{j \in s_M} v_j = w'_M$$

and so, $w_m < w_M$. Therefore, $s_1 \neq s_2 \implies w_1 \neq w_2$.

2.2 CRYPTANALYSIS OF THE BASIC MERKLE-HELLMAN CRYPTOSYSTEM BY A. SHAMIR

In 1984, Adi Shamir presented a polynomial-time algorithm (SHAMIR, 1982) capable of breaking the underlying Merkle-Hellman cryptosystem. The attack exploits a vulnerability that comes from the choice of parameters for the scheme.

Again, let $v_1, \cdots, v_N$ be the secret MH key, i.e., a superincreasing sequence, and $u_1, \cdots, u_N$ the public key, such that

$$u_j \equiv v_{\pi(j)} \theta \pmod{\alpha}, \tag{14}$$

for $j = 1, \cdots, N$, where π is some random permutation and (θ, α) are secret parameters such that $\gcd(\theta, \alpha) = 1$. A reasonable argument shows that a good choice of parameters would be

$$v_1 \approx 2^N \tag{15}$$

and

$$v_N \approx 2^{2N}. \tag{16}$$

In fact, by keeping v_i's approximately within the range from 2^N to 2^{2N}, we seek to achieve both security and efficiency requirements related to the *expansion factor* for the scheme. In terms of efficiency, we are avoiding excessively large values for the v_i's and, consequently, for w_{max}, since the encryption of an n-bit plaintext yields roughly $\log_2 \alpha$ bits of ciphertext. The ciphertext has a maximum number of bits given by $\log_2 n\alpha$ since each term of the public sequence is an integer modulo α, and ciphertext is the sum of at most n of these terms. Hence, the *expansion factor* is given by

$$d = \frac{N}{\log_2 N\alpha}. \tag{17}$$

This number is a measure of how much ciphertext is larger than the corresponding plaintext. Therefore, excessively large values for the v_i's yield a large α, which compromises the efficiency of the system by increasing the expansion factor and, as a consequence, decreasing the information rate (the average number of plaintext bits transmitted per ciphertext bit). Constraining d precludes both low-density attacks (LAGARIAS; ODLYZKO, 1985) and, obviously diminishes the burden on memory and computation. On the other hand, we do not want any v_i's to be too small either, again, for security reasons. Indeed, if we consider the extreme case where $v_1 = 1$, then $u_j = \theta$ for some j, which compromises the system as the value of θ could be revealed after a simple search over N elements.

Under the assumption that both (15) and (16) hold, we may argue that a few first values of the sequence $v_1, \cdots, v_N$ are roughly 2^N. Let us suppose that

$$v_1, \cdots, v_5 \lesssim 2^N. \tag{18}$$

By (14), we may write

$$v_{\pi(j)} \equiv u_j \gamma \pmod{\alpha}, \tag{19}$$

where $\gamma \equiv \theta^{-1} \pmod{\alpha}$. Hence, for all $j = 1, \cdots, N$, there exists an integer k_j such that

$$u_j \gamma - k_j \alpha = v_{\pi(j)}. \tag{20}$$

Dividing both sides by $u_j \alpha$, we obtain

$$\frac{\gamma}{\alpha} - \frac{k_j}{u_j} = \frac{v_{\pi(j)}}{u_j \alpha}. \tag{21}$$

Let $\pi(j_i) = i$, so that $j_i = \pi^{-1}(i)$. Hence, we may write (21) as

$$\frac{\gamma}{\alpha} - \frac{k_{j_i}}{u_{j_i}} = \frac{v_{\pi(j_i)}}{u_{j_i} \alpha}. \tag{22}$$

Under the constraints imposed by (16) and (18), we have that $v_{\pi(j_i)} = v_i \leq 2^N$ and $\alpha > 2^{2N}$ (since, in particular, $\alpha > v_N$). Therefore,

$$\left| \frac{\gamma}{\alpha} - \frac{k_{j_i}}{u_{j_i}} \right| \leq 2^{-3N} \tag{23}$$

for $i = 1, \cdots, 5$. This inequality shows us that $\gamma/\alpha - k_{j_i}/u_{j_i}$ is a small number. More than this, that the fraction $\frac{k_{j_i}}{u_{j_i}}$ is a good approximation for $\frac{\gamma}{\alpha}$. But we need to find the value of k_{j_i}. If we subtract $\gamma/\alpha - k_{j_i}/u_{j_i}$, for $i = 2, \cdots, 5$, from $\gamma/\alpha - k_{j_1}/u_{j_1}$, the inequality still holds, so that

$$\left| \frac{k_{j_i}}{u_{j_i}} - \frac{k_{j_1}}{u_{j_1}} \right| \leq 2^{-3N} \tag{24}$$

for $i = 2, \cdots, 5$. The above inequality yields

$$\frac{|k_{j_i} u_{j_1} - k_{j_1} u_{j_i}|}{|u_{j_1} u_{j_i}|} \leq 2^{-3N}. \tag{25}$$

Provided that $u_{j_i} \leq 2^N$, we finally obtain

$$|k_{j_i} u_{j_1} - k_{j_1} u_{j_i}| \leq 2^N \tag{26}$$

for all $i = 2, \cdots, 5$. After we find the k_{j_i}'s by using integer programming techniques, which can be achieved in polynomial time according to (LENSTRA, 1983), we expect to find an approximation for γ/α which allows us to build a pair (γ', α'), with γ'/α' sufficiently close to γ/α, such that the sequence $w_1, \cdots, w_N$ given by

$$w_j \equiv u_j \gamma' \pmod{\alpha} \tag{27}$$

for $j = 1, \cdots, N$ is superincreasing. If we manage to find such a sequence, we can decipher any message encrypted with the public key $u_1, \cdots, u_N$.

Let us call (γ, α) a *decryption pair*. Shamir basis his attack on the fact that any instance of the Merkle-Hellman cryptosystem has infinitely many decryption pairs. Indeed, any (γ', α'), with γ'/α' sufficiently close to γ/α, will work.

Nevertheless, we must address some issues. The first one is related to finding the indices $j_1, \cdots, j_5$, such that $u_{j_1}, \cdots, u_{j_5}$ correspond to the five smallest terms of the secret knapsack. Since the permutation π is secret, the only way to find the correct indices is by exhaustive search. In other words, the adversary must search among all $\mathcal{O}(N^5)$ possible choices for these indices.

Furthermore, we must add some remarks regarding the number of knapsack elements considered on inequality (18). Recall that we built an integer programming instance, which is given by (26), by guessing the 5 public knapsack items corresponding to the 5 smallest superincreasing elements, assuming that these elements satisfy (18).

There is a reason for this number, but the technicalities behind it are beyond the scope of this work. One can show that the expected number of solutions of the integer program given by (26) is bounded by a constant, which depends precisely on the number of knapsack elements considered, *provided that it is greater than or equal to* 5.

This result is related to the simultaneous Diophantine approximation problem, and further details can be found in (LAGARIAS; HASTAD, 1986). As (LAGARIAS, 1984) demonstrates it, we must choose the number of knapsack elements between the largest of two values, namely 5 or $d' + 2$, where d' is an estimate of the expansion factor defined in (17). Since we are working under the assumption that both (15) and (16) hold, we may consider that the expansion factor is roughly 2 and, consequently, the number of knapsack elements considered for the attack must be 5. In practice, as pointed out by (ODLYZKO, 1990), more than 5 knapsack items may be chosen.

Shamir's attack marked the official break of the original Merkle-Hellman cryptosystem, and the mercy stroke was given by (BRICKELL, 1985), which presented an attack on iterated knapsacks.

2.3 CRYPTANALYSIS OF LAGARIAS AND ODLYZKO

It is worth mentioning that other cryptanalytic techniques, such as the low-density attack proposed by (LAGARIAS; ODLYZKO, 1985), had been successfully used against knapsack-based cryptosystems, which threw severe doubts on the security of such cryptographic constructions.

Equation (28) shows how to reduce the deciphering of MH into an instance of ***s**hortest **v**ector **p**roblem*. There, L is a basis of linearly independent integer vectors, while $\mathbf{x}$ is a coefficient vector representing the (scalar entries of the) plaintext $\mathbf{b}$ to be found, concatenated with entry 1. By design, the correct entries b_i of plaintext $\mathbf{b}$, when applied in $\mathbf{x}$, make the product $L\mathbf{x}$ have the last entry, the only one with absolute value possibly greater then 1, equal 0. δ is a factor yielded by the density d of the parcels and is necessary to ensure the algorithmic solution of SVP for $(L, \mathbf{x})$ would match the cryptanalysis of $\mathbf{b}$. Details go beyond the scope of this work.

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ \delta u_1 & \delta u_2 & \cdots & \delta u_n & -\delta y \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ 1 \end{pmatrix} \tag{28}$$

In the following section, we present a Finite Field Merkle-Hellman algorithm resistant to this cryptanalysis found in later literature.

3 SRVB CRYPTOSYSTEM

Current proposals for knapsack-based asymmetric cryptosystems, as does (WANG; HU, 2010), tend to focus on impeding cryptanalyses based on lattice problems oracles and Diophantine approximations. Therefore, we divide this section into two subsections, each dedicated to a proposal addressing one of these classes of attacks.

3.1 ALTERNATIVE RING

Daniel Santana Rocha proposed a straightforward line of defence specifically for Diophantine approximations-based attacks by substituting the adopted ring from natural integers to a non-ordered one like Gaussian. Essentially, we exploit the isomorphism $\phi : \mathbb{Z}/|\alpha|^2 \to \mathbb{Z}[i]/\alpha$, that happens for $\alpha = a + bi$, with $a, b \in \mathbb{Z}$ being coprime.

We expect the line of defence consisting of the substitution of the ring by a non-ordered one like the *Gaussian integers* is, as well, straightforward: If we tried to replicate Shamir's cryptanalysis *as is* to this new paradigm, we would have to stop precisely between (22) and (23), where the first inequality appears.

It is worth looking in more details into the isomorphism ϕ, which is hard to formalize yet, fortunately, can be easily intuited from an elegant description: imagine a square lattice in the plane of complex numbers, whose side is a hypotenuse of a rectangle triangle of catheti a and b, parallel to the real and imaginary axes. An example of such a lattice is given below. Gaussian integers modulo $\alpha = a + bi$ can be represented by points located within such a lattice. Within each square of such lattice, there are $|\alpha|^2$ (the area of the square) distinct points.

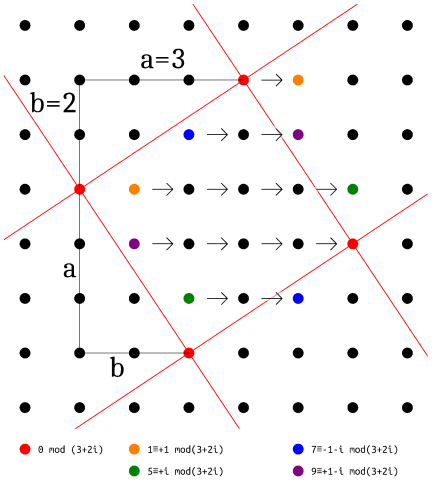


Figure 1 – Lattice on the Argand-Gauss plan.

Figure 1 is the graphic representation of the isomorphism $\phi : \mathbb{Z}/|\alpha|^2 \to \mathbb{Z}[i]/\alpha$, where $\alpha = a+bi = 3+2i$. We highlighted with the same colours the dots of 4 pairs of dots with the same relative position to the red squared lattice. This evinces that if we start at

a red dot and move sequentially to the right, we would perform both a cyclic sequence of 13 different distances to the last red dot roamed (*i.e.,* $y_n = n \pmod{|\alpha^2|}$) and a cyclic sequence of 13 different positions within the red lattice (*i.e.:* $z_n = n \pmod{\alpha}$). ϕ consists simply of mapping the n-th element of the first onto the n-th element of the latter.

Notice that each "jump" from a given row ending on a given color to the other starting with that same color changes the row position r for a given modulus of the lattice to $r \leftarrow (r + b) \mod a + b$ if one counts the rows from top to botton, and, equivalently, to $r \leftarrow (r + a) \mod a + b$ if one counts from bottom up. Hence, the necessary and sufficient condition for each row (and, consequently, each dot) to be roamed exactly once on each cycle is that the size of the jumps is coprime with the number of rows, or, in other words, $\gcd(a, a + b) = \gcd(b, a + b) = 1$, which is equivalent to

$$\gcd(a, b) = 1 \tag{29}$$

The attentive reader could have been troubled by (5) and (29) for suspecting that they might make the keys too scarce or, in other words, the cryptosystem expensive to instantiate, as RSA is. Fortunately, it is not the case: The limit probability of two large random numbers being coprime is as high as $6/\pi^2$ —the inverse of Basel's Constant. Higher than 60%.

Another important practical aspect of this isomorphism is that it roughly preserves memory and computation for arithmetic operations. Notice that it maps a natural integer up to $|\alpha|^2$ to a pair of integers, each with sizes up to about the square root of that, and therefore, have half as many digits. As it will be clear below, in combining the two proposed approaches, all arithmetic operations with public keys are (entry-wise) modular.

3.2 LINEAR ALGEBRA OVER FINITE FIELD

Yuri da Silva Villas Boas proposed a way to hinder cryptanalyses based on oracles for lattice problems, while also reversibly introducing noise to the public message, in a specific way that allows it to be filtered by means of an additional (secret) piece of information (set as an additional part of the private-key). The parameters for this approach are given in Table 2, below:

Table 2 – Parameters for the SRVB

sizing	$((n, N, p), \psi) \in \mathbb{N}^3 \times \mathbb{Z}[i]$
pri key	$((\theta, \alpha), \mathbf{s}, R, U, \pi) \in F^2 \times F^{1\times n} \times F^{n\times n} \times F^{n\times N} \times S_N$
pub key	$P = RU \in \mathbb{F}/p^{n\times N}$
plaintext	$\mathbf{b} \in \{0, 1\}^{N\times 1}$
ciphertext	$\mathbf{z} \in \mathbb{F}/p^{n\times 1}$

Here, N is the block size (as before), p is prime, $F = \mathbb{F}/p$, that is, a **finite field** over p. This is precisely the core of the approach, because the concept of **norm**, that is fundamental for the referred algorithms for SVP in the low density attack, is not applicable to finite fields.

Another thing important to mention is that this approach is perfectly compatible with that of Alternative Ring. In order to implement that, user would have to instantiate a Gaussian integer $\psi = c + di$ having $c^2 + d^2 = p$.

As before, (4) and (5) hold to (θ, α), which, together with π, are instantiated and used in the same ways.

Next, we have

$$\mathbf{e}_1^t U = \mathbf{u}^t, \tag{30}$$

where $\mathbf{e}_1$ is the first canonical vector and $\mathbf{u}$ is the same as in MH. That is, $U_{1,\pi^{-1}(i)}\theta^{-1} = v_i \mod \alpha$ is the ith element of a superincreasing sequence $\mod \alpha$.

For a non-degenerate case (more about that later) we impose:

$$1 < n < N, \tag{31}$$

and

$$rank(R) = rank(U) = n \tag{32}$$

Where $rank(M)$ of a matrix M is the number of linearly independent columns. All entries of random candidates for R and U but those of U's first row are randomly picked from a uniform distribution over $\mathbb{F}/p$.

Accidentally having $rank(P) < n$ would allow an adversary to easily reduce the cryptosystem into one with smaller sizing, given by $(rank(P), N, p)$. This would mean that the instantiating process should check the drawn matrices to ensure (32) holds. Once again, however, the odds are in favor of the defending side and, in fact, much more so. It is easy to see that one so produced candidate R' to R has a probability

$$prob\left(rank\left(R'\right) = n\right) = \prod_{i=1}^{n}(1 - p^{-i}) \tag{33}$$

of being invertible. Since the smallest possible $\mathbf{v}$ is that in which $v_i = 2^{i-1}$, $p \geq \alpha$ and (4) mean that a typical block size $N = 128, 192, 256$ (large enough to make the ciphertext brute-force resistant) would incur in a neglectable probability of a given candidate for R being invalid. Namely, even the largest $n = 127$ for this smallest possible value for $p \approx 2^{128}$ would incur in $prob\left(rank\left(R\right) < n\right) < 10^{-39}$. The probability of a candidate for U with uniformly randomly picked 'noisy entries'

$$prob\left(rank(U) = n\right) = \prod_{i=1}^{n-1}(1 - p^{i-N}) \tag{34}$$

having full rank is even nearer to certainty.

$\mathbf{s}$ is the solution (guaranteed to exist by (32)) for:

$$R^t \mathbf{s}^t = \mathbf{e}_1, \tag{35}$$

Encryption: of plaintext **b** is yielded by:

$$\mathbf{z} = P\mathbf{b} \tag{36}$$

Decryption of ciphertext $\mathbf{z}$ is done by the following 4 steps:

1. compute $y = \mathbf{sz}$

$$y = \mathbf{sz} = \mathbf{s}P\mathbf{b} = \mathbf{s}RU\mathbf{b} = \mathbf{e}_1^t U\mathbf{b} = \mathbf{u}^t\mathbf{b} = \mathbf{u} \cdot \mathbf{b} \tag{37}$$

The attentive reader will, by now, realize that the naming of scalar y, coinciding to original MH's ciphertext was not a consequence of poor judgement since it, in fact, precisely matches the definition used in the original MH. Consequently, the following 3 steps are exactly those of original MH decription.

So, in one sentence, this approach consists of a linear algebraic generalization of MH for n dimensions, specifically based on finite fields, and that allows the ciphertext to be further obfuscated by noise.

$n \geq N$ would mean that:

1. **the system bears** $n - rank(P)$ **dead-weight dimensions** that contribute nothing to security since the system can be easily reduced into one with sizing given by $(rank(P), N, p)$; and possibly
2. $\mathbf{b}$ **can be trivially recovered** using solving a linear system if indeed, as (34) proves it is most likely, $rank(P) = N$;

Finally, $n = 1$ essentially degenerates into the original MH.

3.3 ACHIEVED RESULTS

The proposed schemes achieves:

1. **obliviating order** and, therefore, Shamir's cryptanalysis — Alternative Ring approach;
2. **obliviating norm** and, therefore, lattice-based cryptanalises — Linear Algebraic Generalization approach;
3. **introducing 'noise'** to $\mathbf{u}$ as additional defense;
4. **levelling the size of the (entries of the) ciphertext**, with the benefits described previously;

5. **possibility of eliminating statistical evidence to** α given by the public key ($u_1, \cdots, u_N$ randomly distributed in $\{0, \cdots, \alpha - 1\}$); As a bonus fact, we also have:
6. **simple time and memory complexity and performance analyses**, since the results are mostly the same as those of the equivalent cryptosystem with natural integers times n;

With both Diophantine approximations and Lattice Oracle-based cryptanalyses being precluded, our proposed solution represents a valid contribution to searching for an effective knapsack-based public-key encryption scheme.

An alternative for keeping p private (possibly also accumulating the same role as α) without compromising the uniformity of the size of the sums in the ciphertext most likely would involve a substantial constraint on the variety of the possible sums for each set of parcels. Therefore, it would decrease the information density of d.

4 SECURITY ANALYSIS

We will now present a list discriminating the most effective cryptanalyses, to the best of our knowledge, starting from various components or combinations of components of the private key or a combination of both approaches. Knowledge of public key and ciphertext is assumed.

4.1 NAIVE APPROACH

The naive approach for an instance of this problem with N parcels is, of course, to brute-force attempt up to all 2^N possibilities, with time complexity $\mathcal{O}(2^N)$ and space complexity of $\mathcal{O}(N)$;

4.2 MEET-IN-THE-MIDDLE, BY HOROWITZ AND SAHNI

Next, (HOROWITZ; SAHNI, 1974) devised a *meet-in-the-middle* approach in which the N parcels are arbitrarily subdivided in two halves H and L (if N is odd, then one of the partitions closest to equality of cardinalities). For each of them, all the (approximately) $2^{\frac{N}{2}}$ subsums are pre-calculated and sorted. The algorithm proceeds to try out combinations of one sum from each half, starting from the highest in H them and the lowest in L. If the currently selected parcels sum up more than the desired value y, the algorithm advances one step down in H, and, conversely, if the current sum surpasses y, the algorithm advances one step up in L. Time and space complexity of $\mathcal{O}(2^{\frac{N}{2}})$;

4.3 IMPROVED MEET-IN-THE-MIDDLE, BY SCHROEPPEL AND SHAMIR

(SCHROEPPEL, 1981) improved on (HOROWITZ; SAHNI, 1974) by further dividing each of two halves L and H described just above in halves again (or, like before, the closest possible to it), say, L_1 and L_2; and H_1 and H_2 respectively. The idea is that only the $\mathcal{O}(2^{\frac{N}{4}})$ subset sums of the 'fourths' L_1, L_2, R_1 and R_2, rather than the $\mathcal{O}(2^{\frac{N}{2}})$ ones of the 'halves' L and R are pre-calculated, kept in memory and ordered. The algorithm them proceeds to emulate the previous one by calculating subset sums of L and R on demand from L_1 and L_2; and R_1 and R_2 respectively, in each the $\mathcal{O}(2^{\frac{N}{2}})$ steps of an adaptation of the meet-in-the-middle elimination process.

4.4 CLASSICAL AND PROBABILISTIC ALGORITHM BY ANJA BECKER, JEAN-SÉBASTIEN CORON, AND ANTOINE JOUX

(BECKER et al., 2011) designed a classical probabilistic algorithm with even lower expected complexity. Its main idea is to represent the vectors that multiply the

public key by the sum of two others belonging to a family of pairs purposefully not provided with unicity (that is, more than one pair represent the exact plain text). Each of these two parcels is recursively represented the same way until a recursion depth of 4. It is precisely the multiplicity of said representations that causes the probability of each attempted verification of possible solution to increase. It can be proved that in all but statistically negligible pathological cases, the algorithm works with time complexity bounded by $\mathcal{O}(2^{0.291N})$. Details go beyond the scope of this work.

4.5 QUANTUM ALGORITHMS

"In 2013, (BERNSTEIN et al., 2013) constructed quantum subset sum algorithms, inspired by the classical algorithms above. Namely, Bernstein et al. showed that quantum algorithms for the naive and Meet-in-the-Middle approach achieve run time $2^{\frac{n}{2}}$ *and* $2^{\frac{n}{3}}$*, respectively. Moreover, a first quantum version of Schroeppel-Shamir with Grover search (GROVER, 1996) runs in time* $2^{\frac{3n}{8}}$ *using only space* $2^{\frac{n}{8}}$*. A second quantum version of Schroeppel-Shamir using quantum walks (AHARONOV et al., 1996) and (AMBAINIS, 2007) achieves time* $2^{0.3n}$*. Eventually, Bernstein, Jeffery, Lange, and Meurer used the quantum walk framework of Magniez et al. (MAGNIEZ et al., 2011) to achieve a quantum version of the Howgrave-Graham, Joux algorithm with time and space complexity* $2^{0.241N}$*."*

Verbatim from (HELM; MAY, 2018). Finally, the same (HELM; MAY, 2018) proposed a quantum algorithm with a time complexity of $2^{0.226N}$. Details go beyond the scope of this work.

4.6 θ OR α

Knowledge of θ or α alone is unknown to enable and help for cryptanalysis to SRVB.

4.6.1 *n* columns of *U*

Finding out n columns of U enables an algorithm for deriving R, which can subsequently be used to reduce full SRVB into only the first approach (see below). The method is very straightforward: the i-th column of U, U^i, yields the i-th column of P, P^i, when multiplied in the left by R. This allows for the equation

$$R\begin{pmatrix} U^{i_1} & \cdots & U^{i_n} \end{pmatrix} = \begin{pmatrix} P^{i_1} & \cdots & P^{i_n} \end{pmatrix}, \tag{38}$$

where $i_j \in 1, \cdots, n$ is the j-th column index of the resulting square matrices made up of n columns of U and P. That trivially provides R by means of n instances of n by n linear systems. In case the indexes i_j's are not known, the attacker would still have to

brute force check amongst all $\frac{N!}{(N-n)!}$ possible mappings M: $\{1, \cdots, n\} \longrightarrow \{1, \cdots, N\}$. Total complexity would therefore be $O(\frac{N!}{(N-n)!} n^4 l)$.

4.6.2 R or $\mathbf{s}$

R allows for trivially finding R^{-1}, and then $U = R^{-1}P$, and $\mathbf{s} = (R^{-1}\mathbf{e}_1)^t$. At this point, an attacker can obtain, like before,

$$y = \mathbf{sz} = \mathbf{s}RU\mathbf{b} = \mathbf{u} \cdot \mathbf{b} \tag{39}$$

From this point on, the attacker has the same problem as they would have to cryptanalyse SRVB with the alternative ring approach only.

This preliminary analysis shows it is crucial to ensure brute-force resistance of s, R and (sets of n) U's columns, which, measured in bits of entropy, equals nl, $n^2 l$ and $(n)nl$ respectively, where $l \approx log_2(p)$ is the length of entries. On the other hand, the possibility of the adversary performing an (incomplete) Gaussian elimination in P yields an effective ciphertext brute-force resistance of only $(N-n)l$. This means that we must have

$$n \leq N/2 \tag{40}$$

since increasing n beyond that would both increase resource consumption and weaken security.

Although a preliminary analysis suggests that the linear algebraic generalization approach layer could actually substitute that of MH, we will, at this moment, keep the rationale behind (8) and, for good measure, require at least $v_1 \approx 2^{128}$. This yields the suggested sizing:

$$2 \leq n \leq N/2 = 128 = l/3 \tag{41}$$

As explained before, complexity and performance analysis of SRVB will mostly equal n times those of MH with the same sizing, except for decryption, which has the multiplication by $\mathbf{s}$ ($O(nl)$ time and memory-wise) more.

Further research is still needed to assess provable security for the proposed cryptosystem, like many other knapsack-based or otherwise public-key cryptosystems.

5 FINAL REMARKS

Despite definitive refutation of the original MH and many of its variants and the persistence of distrust by the public on knapsack-based asymmetric cryptosystems, new attempts to revive it continue to appear. Reasons for it include:

1. **very high complexity asymmetry**: $\mathcal{O}(N)$ for encryption and decryption versus, on the S4P component alone, $\mathcal{O}(2^{N/2})$ for best known classical, deterministic attack, $\mathcal{O}(2^{0.291N})$ for best classical attack, and $\mathcal{O}(2^{0.226N})$ for the best quantum attack; That yields
2. **high effectiveness** of the cryptosystem if and while cryptanalyses are not discovered;
3. **excellent performance** time and memory-wise;
4. **exercising the concept of homomorphic cryptosystem** it uses as a component; while also keeping
5. **elegant simplicity**: requiring elementary level mathematics to be understood, implemented, and used;

Our contribution to knapsack-based cryptosystems successfully prevents both Diophantine approximations and lattice-oracle-based attacks at the same time. Even though security of the proposed cryptosystem is left unproved, our research already does contribute to addressing the still open question of whether or not knapsack-based asymmetric encryption schemes have any future. Successful cryptanalysis to SRVB is being offered a US$2560,00 (one thousand *Knuth Dollars*) prize in the Nep.Sec Contest. If and when the contest is won, SRVB happens to be proved to be ineffective (as so many others have), it will, at least, be an elementary school–accessible material of diffusion for cryptology.

Further works may involve or investigate possibility of:

1. **Detailing** complexity analysis;
2. **Implementing SRVB** in high and low level languages;
3. **Programming Nep.Sec Contest** as a smart contract in a blockchain, therefore turning it in a 0-trust, no-questions-asked contest;
4. **Assessing Post-quantumness**;
5. **Adding a signature scheme** based on **C**losest **V**ector **P**roblem, like GGH or NTRUE;

6. **Generalizing the Alternative Ring Approach** for quartenions, or **changing** it to Eiseinstein integers;
7. **Generalizaing the Finite Fields Approach** from primes to polynomials with degree greater than 0;
8. **Demonstrating** security.

REFERENCES

AHARONOV, Dorit; AMBAINIS, Andris; KEMPE, Julia; VAZIRANI, Umesh. Quantum walks on graphs. **In Proceedings of the thirty-third annual ACM symposium on Theory of computing**, ACM, p. 50–59, 1996.

AMBAINIS, Andris. Quantum walk algorithm for element distinctness. **J. Comput.**, SIAM, p. 210–239, 2007. DOI: 10.1137/S0097539705447311. Available from: http://dx.doi.org/10.1137/S0097539705447311.

BECKER, Anja; CORON, Jean-Sébastien; JOUX, Antoine. Improved generic algorithms for hard knapsacks. **In Annual International Conference on the Theory and Applications of Cryptographic Techniques**, Springer, p. 364–385, 2011.

BERNSTEIN, Daniel J; JEFFERY, Stacey; LANGE, Tanja; MEURER, Alexander. Quantum algorithms for the subset-sum problem. **International Workshop on Post-Quantum Cryptography**, Springer, p. 16–33, 2013.

BRICKELL, Ernest F. Breaking Iterated Knapsacks. In: **Advances in Cryptology: Proceedings of CRYPTO 84**. Ed. by George Robert Blakley and David Chaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985. P. 342–358. ISBN 978-3-540-39568-3. DOI: 10.1007/3-540-39568-7_27. Available from: https://doi.org/10.1007/3-540-39568-7_27.

DIFFIE, W.; HELLMAN, M. New Directions in Cryptography. **IEEE Trans. Inf. Theor.**, IEEE Press, Piscataway, NJ, USA, v. 22, n. 6, p. 644–654, Nov. 1976. ISSN 0018-9448. DOI: 10.1109/TIT.1976.1055638. Available from: http://dx.doi.org/10.1109/TIT.1976.1055638.

GROVER, Lov K. A fast quantum mechanical algorithm for database search. **In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing**, ACM, p. 212–219, 1996.

HELM, Alexander; MAY, Alexander. Improved generic algorithms for hard knapsacks. **13th Conference on the Theory of Quantum Computation, Communication and Cryptography**, Dagstuhl, p. 364–385, 2018.

HOROWITZ, Ellis; SAHNI, Sartaj. Computing Partitions with Applications to the Knapsack Problem. **J. ACM**, ACM, New York, NY, USA, v. 21, n. 2, p. 277–292, Apr.

1974. ISSN 0004-5411. DOI: 10.1145/321812.321823. Available from: http://doi.acm.org/10.1145/321812.321823.

LAGARIAS, J. C. Performance analysis of Shamir's attack on the basic Merkle-Hellman knapsack cryptosystem. In: **Automata, Languages and Programming: 11th Colloquium Antwerp, Belgium, July 16–20, 1984**. Ed. by Jan Paredaens. [S.l.: s.n.], 1984. P. 312–323. ISBN 978-3-540-38886-9. DOI: 10.1007/3-540-13345-3_28. Available from: https://doi.org/10.1007/3-540-13345-3_28.

LAGARIAS, J. C.; HASTAD, J. Simultaneous Diophantine approximation of rationals by rationals. **J. Number Theory**, n. 24, p. 200–228, 1986.

LAGARIAS, J. C.; ODLYZKO, A. M. Solving Low-density Subset Sum Problems. **J. ACM**, ACM, New York, NY, USA, v. 32, n. 1, p. 229–246, Jan. 1985. ISSN 0004-5411. DOI: 10.1145/2455.2461. Available from: http://doi.acm.org/10.1145/2455.2461.

LENSTRA, H. W. Integer Programming with a Fixed Number of Variables. **Mathematics of Operations Research**, INFORMS, v. 8, n. 4, p. 538–548, 1983. ISSN 0364765X, 15265471. Available from: http://www.jstor.org/stable/3689168.

MAGNIEZ, Frédéric; NAYAK, Ashwin; ROLAND, Jérémie; SANTHA, Miklos. Search via quantum walk. **Journal on Computing**, SIAM, p. 142–164, 2011.

MERKLE, R.; HELLMAN, M. Hiding Information and Signatures in Trapdoor Knapsacks. **IEEE Trans. Inf. Theor.**, IEEE Press, Piscataway, NJ, USA, v. 24, n. 5, p. 525–530, Sept. 1978. ISSN 0018-9448. DOI: 10.1109/TIT.1978.1055927. Available from: http://dx.doi.org/10.1109/TIT.1978.1055927.

ODLYZKO, A. M. The rise and fall of knapsack cryptosystems. In: IN Cryptology and Computational Number Theory. [S.l.]: A.M.S, 1990. P. 75–88.

SCHROEPPEL, Adi Shamir. A t=o(2^n/2), s=o(2^n/4) algorithm for certain np-complete problems. **journal on Computing**, SIAM, v. 10, n. 3, p. 456–464, 1981. ISSN 0001-0782. DOI: 10.1145/359340.359342. Available from: http://doi.acm.org/10.1145/359340.359342.

SHAMIR, Adi. A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. In: (SFCS '82), p. 145–152. DOI: 10.1109/SFCS.1982.55. Available from: http://dx.doi.org/10.1109/SFCS.1982.55.

THANGAVEL, M.; VARALAKSHMI, P. A novel public key cryptosystem based on Merkle-Hellman Knapsack Cryptosystem. In: 2016 ICoAC). [S.l.: s.n.], Jan. 2017. P. 117–122. DOI: 10.1109/ICoAC.2017.7951756.

WANG, Baocang; HU, Yupu. Quadratic compact knapsack public-key cryptosystem. **Computers & Mathematics with Applications**, Elsevier, p. 194–206, 1 2010. ISSN 0898-1221.