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

Lucas Pandfoldo Perin

Message Encoding Algorithms for Winternitz Signatures

Florianópolis
2021

Lucas Pandfoldo Perin

Message Encoding Algorithms for Winternitz Signatures

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

Florianópolis
2021

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

Perin, Lucas Pandolfo
Message Encoding Algorithms for Winternitz Signatures / Lucas Pandolfo Perin ; orientador, Ricardo Felipe Custódio, coorientador, Daniel Panario, 2021.
106 p.

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

Inclui referências.

1. Ciência da Computação. 2. Criptografia. 3. Assinaturas Digitais. 4. Assinaturas Baseadas em Hash. 5. Criptografia pós-quântica. I. Custódio, Ricardo Felipe. II. Panario, Daniel. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Lucas Pandfoldo Perin

Message Encoding Algorithms for Winternitz Signatures

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

Prof^{a}. Thaís Bardini Idalino, Dr.
Universidade Federal de Santa Catarina

Prof. Marco Aurélio Amaral Henriques, Dr.
Universidade Estadual de Campinas

Prof. Jeroen van de Graaf, Dr.
Universidade Federal de Minas Gerais

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

ICP-Edu
Documento assinado digitalmente
Vania Bogorny
Data: 11/11/2021 08:40:44-0300
CPF: 684.586.570-15
Verifique as assinaturas em https://v.ufsc.br

Prof. Vania Bogorny, Dr.
Coordenadora do Programa

ICP-Edu
Documento assinado digitalmente
Ricardo Felipe Custodio
Data: 10/11/2021 08:28:27-0300
CPF: 485.981.609-97
Verifique as assinaturas em https://v.ufsc.br

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis, 2021.

To mom and dad.

ACKNOWLEDGEMENTS

There is a long list of people which I would like to personally thank, for all the support that they gave me during this long period pursuing my PhD. While I cannot list every single one of them, hopefully I can include most in this short page.

I have to start at my academic origin, where I spent most of my days since 2012, the security laboratory at UFSC — LabSEC. I would like to thank the many friends, colleagues and professors, that have somehow been there with me and for me, during this long time at LabSEC. I thank Lucila Alosilla, for always helping our team and for taking good care of us. I also thank my coauthors Prof. Jean Martina, Gustavo Zambonin and Douglas Martins that help me so much kick starting the research on post-quantum signatures.

I would also like to extend my gratitude to the TII family. One can only wish to work with such great minds and great people. I am grateful to be part of the team, and look forward for the new experiences.

A special thank you goes to “Team Canada”, without whom this thesis would not have been possible. Thank you to Prof. Lucia Moura and my co-supervisor Prof. Daniel Panario, for the uncountable hours you have spent working with me. Your dedication and friendship are an inspiration.

Another special thanks goes to my supervisor, Prof. Ricardo Custódio. For over 7 years under your supervision during my Master degree and the PhD, you have taught me so much. I appreciate all the opportunities that you have given me and all of the things that we have accomplished together. Thank you.

On a more personal note, I would like to thank my dear family. During such hard times, I am blessed with loved ones that have supported me through thick and thin. Thank you to my brother for always pushing me to greater heights, and thank you to my parents for giving me all the things I need and, sometimes, the things I desire. At last, I extend my most sincere gratitude to my beautiful wife and her family. Carla, I am so grateful to you and for all the times you stood by me. Sorry it so took long!

Never tell me the odds! — Han Solo

RESUMO

Considerando um determinado conjunto de parâmetros para o esquema de assinatura única de Winternitz (WOTS), a complexidade total da geração e verificação de uma assinatura é constante e independente do documento a ser assinado. Esses custos são devidos ao número de iterações de uma função f, executados sobre elementos de uma chave privada. No entanto, o custo de geração de assinatura por si só pode ser diferente do custo da verificação de assinatura, dependendo diretamente do documento de entrada. Este trabalho apresenta uma nova variante do esquema WOTS, permitindo o ajuste desses custos. Ou seja, aumenta-se o tempo de geração de assinatura em favor de uma verificação mais rápida ou vice-versa. O número total de repetições de f para parâmetros específicos do esquema podem ser reduzidos, ocasionando também uma redução do custo de geração de chaves. Na contribuição principal deste trabalho, permite-se escolher um custo fixo de execuções de f, inalterado para qualquer mensagem de entrada. Experimentos mostram que as propostas têm impacto substancial em esquemas de assinatura baseados em árvores de Merkle, como XMSS. Além disso, se f for uma função de direção única, resistente à segunda pré-imagem e indetectável, prova-se formalmente que o esquema é *Existentially Unforgeable under a Chosen Message Attack* (EU-CMA).

Palavras-chave: Assinaturas baseadas em funções Hash. Winternitz. Criptografia. Criptografia Pós-Quântica. Análise Combinatória.

RESUMO ESTENDIDO

Introdução

A criptografia clássica atingiu seu ápice com o desenvolvimento de máquinas de rotor, como a famosa máquina enigma, recentemente retratada no filme "O Jogo da Imitação". Diante disto, testemunhou-se uma mudança de paradigma com o advento dos computadores pessoais, então nasceu a criptografia moderna. O Data Encryption Standard (DES), desenvolvido no início dos anos 70, é um marco importante no história da criptografia. No entanto, e talvez mais importante, a forma como usamos criptografia no mundo moderno mudou completamente em 1976 quando a Criptografia de Chaves Públicas (CCP) foi proposta pela primeira vez por Whitfield Diffie e Martin Hellman em seu artigo intitulado “New directions in cryptography" (DIFFIE; HELLMAN, 1976).

Possivelmente vivemos um momento de recorrência da história, pois estamos prestes a testemunhar a criação de computadores quânticos; computadores com tal poder que ameaça quebrar a maioria dos sistemas criptográficos usados por bilhões de usuários na Internet diariamente. Muitas perguntas sem resposta ainda estão em jogo: Algum dia teremos um computador quântico prático e de grande escala? Se for assim, teria ele a capacidade computacional comparável às expectativas teóricas? Ou talvez, para aqueles em favor das teorias da conspiração: e se um computador quântico já existir e não estivermos cientes dele? Essas questões têm atraído a atenção da comunidade internacional, clamando por soluções à medida que entramos nessa nova era de criptografia.

Nesta tese, estuda-se um dos candidatos mais antigos do que hoje chamamos de Criptografia Pós-Quântica (CPQ). Ou seja, sistemas criptográficos que podem ser utilizados por computadores que temos disponíveis hoje e que são considerados seguros mesmo contra adversários quânticos. É o caso de Assinaturas Baseadas em *Hash* (ABH), os quais são esquemas com reduções de segurança bem conhecidas e considerados seguros contra computadores quânticos. Esses esquemas usam apenas criptografia simétrica, mais especificamente funções de *Hash* criptográfico, que tem sido estudadas há décadas. Por esse motivo, há pouca preocupação quanto ao amadurecimento do seu atual estado da arte. No entanto, tem seus desafios, já que ABH resolve o problema de assinaturas digitais no cenário quântico introduzindo suas próprias limitações.

Um dos pilares de ABH é o esquema de assinatura única de Winternitz WOTS (MERKLE, 1989). Como seu nome sugere, ele tem a capacidade de realizar uma única assinatura. No entanto, instâncias distintas do mesmo esquema de ABH podem ser agrupadas em esquemas mais complexos para se obter esquemas de múltiplas assinaturas. Por exemplo, os esquemas de assinaturas múltiplas XMSS e LMS, com padrões públicos RFC 8391 (HÜLSING et al., 2018) e RFC 8554 (MCGREW; CURCIO; FLUHRER, 2019), respectivamente. Infelizmente, essas soluções introduzem uma condição de estado, isto é, o estado das chaves privadas agregadas a uma árvore Merkle deve ser mantido com segurança. Consequentemente, concluiu-se que o uso de padrões de ABH com estado só deve ser considerado para certas aplicações onde a integridade do estado da chave privada pode ser garantida (COOPER et al., 2020). Ademais, um esquema de ABH mais complexo, que remove a restrição de estado das chaves, foi submetido à “competição[1]” de padronização de CPQ da instituição Americana National Institute of Standards and Technology (NIST) atualmente em andamento (ALAGIC et al., 2020). Este esquema é conhecido como SPHINCS+ (BERNSTEIN et al., 2019; AUMASSON et al., 2020) e está sendo considerado como candidato alternativo na terceira rodada do processo de submissão.

Um aspecto interessante de ABH é como a distribuição de bits da mensagem a ser assinada pode

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

afetar o desempenho de alguns esquemas. Por exemplo, o número de operações hash executadas nas etapas de assinatura e verificação do WOTS é determinado diretamente pela mensagem de entrada. Durante essas etapas, o esquema usa um algoritmo de codificação que gera uma tupla de elementos que representam exclusivamente a mensagem. Os elementos dessa tupla são então usados para determinar o número de operações hash em cada etapa. Esta característica foi explorada por (STEINWANDT; VILLÁNYI, 2008) e mais recentemente por (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018) e (ROH; JUNG; KWON, 2018). Esses trabalhos propõem diferentes maneiras de codificar uma mensagem, de forma que o processo de assinatura ou o processo de verificação (não ambos) utilizem uma quantidade menor de execuções de uma função hash. Ao fazer isso, outros desafios surgem. Em geral, espera-se que, ao reduzir o número de funções hash executadas na etapa de verificação, mais funções hash tenham que ser executadas na etapa de assinatura ou vice-versa. De fato, todos os trabalhos relacionados na literatura propõem tal compensação, além de um custo geral aumentado para todo o esquema. Ou seja, a geração de chaves também é afetada e, portanto, mais operações de hash também devem ser realizadas nesta etapa.

Neste trabalho levantam-se as seguintes questões: será possível explorar funções de codificação alternativas que não aumentem o custo de geração de chaves e, ao mesmo tempo, permitam tal compensação? Existem algoritmos de codificação que podem reduzir o custo de geração de chaves, ou seja, reduzir todo o custo do esquema, diminuindo o número de funções hash necessárias para gerar chaves, assinar e verificar assinaturas, ao mesmo tempo? É possível realizar isso com eficiência? Por último, e talvez mais importante, é possível fazer isso preservando a segurança, em comparação com os esquemas de estado da arte na literatura?

Objetivos

O esquema de assinatura de Winternitz utiliza um parâmetro denominado w, que permite uma compensação clara entre desempenho e tamanho de assinatura. No entanto, o desempenho relacionado ao número de funções hash necessárias para produzir ou verificar uma assinatura Winternitz também é afetado pela mensagem a ser assinada. O principal objetivo desta tese é ampliar nosso conhecimento sobre como as mensagens impactam o desempenho do esquema de assinatura única de Winternitz e como podemos usar isso em nossa vantagem para diferentes cenários de aplicação. Neste trabalho, serão avaliados os diferentes tipos de codificação de mensagens que podem ser usados com Winternitz One Time Signature (WOTS) e como eles o afetam. Como resultado, pretende-se desenvolver e melhorar novas técnicas que possam ser utilizadas para reduzir o custo associado à geração de assinaturas, verificação de assinaturas e geração de chaves do esquema baseado em WOTS. Como principal objetivo, serão desenvolvidos algoritmos eficientes e com tais propriedades. Também serão feitas avaliações de seu desempenho na prática, assim como a prova de segurança do esquema proposto, a fim obter-se níveis de segurança comparáveis ao estado da arte.

Motivação

Recentemente, NIST publicou uma chamada para candidatos à padronização de algoritmo de criptografia pós-quântica (NIST, 2016). Ademais, nota-se que fabricantes de chips já alcançaram grandes avanços no que poderia ser o próximo passo para a revolução da informática, com os Computadores Quânticos. Enquanto isso, a criptografia pós-quântica ainda está em seus primeiros estágios de desenvolvimento, onde as discussões sobre segurança e melhorias de desempenho ainda são constantemente debatidas em fóruns públicos. Portanto, há muita motivação e interesse na comunidade acadêmica internacional para pesquisas neste escopo, tornando o momento para contribuir com um tema dessa envergadura bastante oportuno.

Considerando o estado atual de CPQ, os candidatos à ABH são possivelmente as alternativas mais proeminentes quando se considera a implantação imediata de esquemas de assinatura pós-quânticos no mundo real. Embora a maturidade e a segurança desses esquemas já sejam bem conhecidas, deseja-se buscar novas alternativas que ainda não foram exaustivamente analisadas na literatura.

Limitações do trabalho

Existem diversas alternativas para esquemas ABH no estado atual da literatura. Além disso, parece que um caso de uso comum para esquemas assinatura única é agregação de múltiplas instâncias destes em Árvores Merkle, afim de construir esquemas de múltiplas assinaturas, sob a mesma chave pública. Outros aplicativos interessantes que são comumente referidos no processo de padronização da instituição NIST são implementações para dispositivos restritos, como o Cortex-M4.

Portanto, propõe-se um estreitamento do escopo, para que se possa avaliar as contribuições dentro deste trabalho. Serão abordados apenas os algoritmos de codificação para esquemas baseados em WOTS. Como cenário de aplicação, serão avaliados experimentos usando o código de referência do esquema EXtended Merkle Signature Scheme (XMSS), que se ajusta bem para demonstrar a capacidade das contribuições aqui propostas, mas sem fornecer resultados para outros esquemas de múltiplas assinatura.

O desempenho dos algoritmos propostos serão abordados, assim como novas técnicas para reduzir seu tempo de execução. No entanto, não serão discutidas em detalhes otimizações de código, pois espera-se que os tetes de desempenho já demonstrem as limitações existente e sugeridas para trabalhos futuros. Além disso, existem inúmeros desafios para implementar algumas das contribuições em dispositivos embarcados. Portanto, embora não serão discutidos meios de implementar os algoritmos propostos em tais plataformas, serão abordadas alternativas de como contornar essas limitações para um conjunto específico de cenários de aplicativos.

Método

Para esta tese, não serão utilizados métodos de pesquisa qualitativa ou quantitativa com o objetivo de reunir e filtrar trabalhos relacionados da literatura. Pode-se facilmente resumir o estado da literatura, abordando cada trabalho individualmente, visto que poucos trabalhos na abordam o tema discutido. O problema da codificação de inteiros, por outro lado, está em um nível diferente. Esse é um problema clássico em Ciência da Computação e Matemática, e não pretende-se fazer um levantamento e revisão deste tema no escopo de desse trabalho.

A pesquisa será conduzida investigando a literatura e enumerando propriedades distintas que são interessantes para ABH. Serão avaliados principalmente os ganhos de compensação do uso de diferentes algoritmos de codificação em relação ao número de funções hash que são executadas no esquema de assinatura. Ou seja, o número médio ou exato de funções hash executadas para assinar ou verificar uma mensagem ao usar funções de codificação distintas será a principal métrica de avaliação. Para fornecer comparação justa, serão considerados tamanho de assinatura semelhante ou igual para cada esquema proposto e no mesmo nível de segurança apresentado pelo estado da arte.

Finalmente, a base matemática necessária para construir adequadamente as propostas é fornecida. Para validar o trabalho, são detalhadas estimativas e/ou resultados exatos dos custos e que podem ser usados para comparação com outros métodos. Além disso, serão conduzidos experimentos que comparam o tempo de execução das propostas da tese com as alternativas de codificação em relação a trabalhos anteriores, afim de abordar com mais detalhe a questão da eficiência.

Principais Contribuições

As principais contribuições deste trabalho estão relacionadas diretamente à ABH e publicadas em (PERIN et al., 2018) e (PERIN et al., 2021). O primeiro trabalho é o resultado de um estudo preliminar de ABH. Analisa-se como mensagens afetam o número de operações de hash avaliadas nas etapas de assinatura e verificação do esquema WOTS e, em seguida, é proposto uma técnica para aleatorizar essa mensagem, de modo que a assinatura ou a verificação (não ambos) possam ser realizadas com menos iterações de hash.

O último é o resultado do objetivo principal desta tese. Observa-se propriedades interessantes em uma função de codificação de soma constante e proposta em (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018). O esquema é atualizado para que seu nível de segurança fique no mesmo patamar que o esquema Winternitz One Time Signature Plus (WOTS+), para o qual uma prova completa e formal é dada. Além disso, são propostas diversas melhorias relacionadas à função de codificação, um algoritmo determinístico e parâmetros que apresentam melhor desempenho geral, diminuindo o custo de geração de chaves do nosso esquema.

Outras contribuições

Durante o curso de desenvolvimento deste trabalho, também foram abordados outras temas de pesquisa acadêmicas. Embora esses temas não estejam exatamente alinhados com o escopo desta tese, segue um breve resumo desses trabalhos. A menção destas contribuições é relevante para transmitir as atividades de colaboração com pesquisadores renomados e internacionais em assuntos multidisciplinares.

É interessante compreender as permutações geradas pela avaliação da função $g^i \pmod{p}$, para $i = 1, \dots p - 1$, com p primo e onde g é um elemento primitivo de um grupo multiplicativo inteiros módulo p. Em outras palavras, g^i gera o conjunto $\{1, \dots, p - 1\}$ e escolhas distintas de g para um p fixo produzem permutações distintas do mesmo conjunto.

Este é um problema interessante, dado que a exponenciação de elementos primitivos de grupos multiplicativos de ordem prima desempenha um papel importante em primitivas criptográficas como, por exemplo, no esquema de assinatura ElGamal (ELGAMAL, 1985). Foram estudadas e publicadas evidências (NIEHUES et al., 2020) de que essa função gera permutações que preservam propriedades esperadas de permutações aleatórias.

Além disso, deu-se continuidade a este estudo através de uma abordagem diferente, produzindo sequências a partir dessas permutações. São chamadas de *Sequências ElGamal* (PANARIO; PERIN; STEVENS, N.D.). A sequência é bastante simples, para alguma permutação $\pi = \{\pi_i : g^i \pmod{p},\ 1 \le i < p\}$, denota-se uma Sequência ElGamal para algum $v \mid p - 1$ como $\sigma = \{\sigma_i : \pi_i \pmod{v}\}$.

Neste artigo, são detalhados limites teóricos e experimentais do número de ocorreências consecutivas e tuplas em sequências ElGamal. Também são detalhados os limites para sequências geradas a partir de uma permutação aleatória, seguido por uma discussão entre a relação entre ambos.

Resultados e Considerações Finais

Nesta tese, foram revisadas as alternativas de codificação que foram propostas para substituir a codificação `base-w` originalmente utilizada em WOTS e sua variante estado da arte WOTS+. Propôs-se duas contribuições principais, a saber WOTS-BR e WOTS-CS+. O primeiro é uma composição de duas técnicas simples que produzem resultados notáveis para melhorar o tempo de execução de verificação de assinatura com uma compensação em que o tempo de execução de geração de assinatura é aumentado. O segundo é uma melhoria da alternativa de codificação de soma constante, que minimiza o custo de geração de chave para esquemas baseados em

Winternitz, reduzindo assim a assinatura e verificação também dos tempos de execução.

Para WOTS-BR, foi apresentado um preenchimento (*padding*) na codificação `base-w` que permite uma verificação mais rápida para WOTS+ e, consequentemente, XMSS. Esta proposta não é compatível com RFC 8391, mas espera-se que seja considerada em caso de futuras revisões de padrões de assinatura baseados em hash. Além disso, uma técnica para aleatorizar a mensagem a ser assinada que pode aumentar consideravelmente o desempenho da verificação de assinatura foi proposta. Esta técnica foi posteriormente adaptada para ser compatível com RFC 8391 e mostrou reduzir o custo médio de verificação de uma única assinatura WOTS+ em até $55,5\%$ (BOS et al., 2020, Tabela 7.2). Os mesmos autores utilizam nossa contribuição, juntamente com outras modificações presentes na literatura, para atingir a verificação de XMSS do tempo de execução em dispositivos Cortex-M4 na metade do tempo médio esperado.

Para o caso de WOTS-CS+, foram apresentados novos algoritmos de codificação determinística para a variante de soma constante do esquema de assinatura de tempo único de Winternitz, WOTS-CS. Esses métodos reduziram os custos associados à função de codificação, empregando técnicas distintas, como programação dinâmica e busca binária. Não apenas alcançou-se uma codificação mais rápida do que a alternativa probabilística da literatura, mas também foram expandidos os conjuntos de parâmetros que podem ser usados com WOTS-CS+, visando reduzir o custo de geração de chaves para assinaturas baseadas em hash.

Como resultado, a abordagem de soma constante permite um trade-off flexível entre os custos de geração de chave, geração e verificação de assinatura, aceitando ambas as estratégias de seleção de parâmetros: MINVER ou MINGEN. Essa abordagem permite reduzir o número de aplicações da função de encadeamento para a etapa de geração de chaves, alcançando custos melhores e mais competitivos aos obtidos com WOTS+. Observa-se que reduzir o custo de geração de chave é particularmente interessante, pois diminui o custo geral de todas as etapas relacionadas à assinatura e não é possível por outros trabalhos relacionados à soma constante (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018).

Alternativamente, são propostos parâmetros distintos que podem ser usados com aplicações específicas, reduzindo o custo de geração de assinatura em vez de verificação. Este é um cenário relevante para muitos casos relacionados à assinatura de documentos eletrônicos. Dispositivos de assinatura podem tirar vantagem do custo fixo, previsível e reduzido de assinatura sem ter que lidar com o custo de codificação.

Outra contribuição significativa é o estudo abrangente das tuplas de soma constante. São provadas propriedades interessantes e fundamentais para a análise de segurança do esquema. Sob uma ampla gama de parâmetros aceitáveis, é provado que WOTS-CS+ é Existentially Unforgeable under a Chosen Message Attack (EU-CMA) desde que $\mathcal{F}_\lambda$ seja uma família de função de direção única, indetectável e resistente à segunda pré-imagem. Esta é uma grande melhoria sobre (KAJI; CRUZ; YATANI, 2018), colocando WOTS-CS+ no mesmo patamar que WOTS+ para os mesmos tamanhos de assinatura. Anteriormente, o esquema de segurança de soma constante dependia de suposições mais fortes, como resistência à colisão.

Palavras-chave: Assinaturas baseadas em funções Hash. Winternitz. Criptografia. Criptografia Pós-Quântica. Análise Combinatória.

ABSTRACT

It is known that, for a given set of parameters, the overall complexity for generating and verifying a signature is constant and independent of the document being signed, for the Winternitz one-time signature scheme (WOTS). These costs are due to the number of chained iterations of a function f. However, the cost for signature generation alone is slightly different from signature verification, and these depend on the message to be signed. We introduce a new variant for WOTS, which allows the adjustment of these costs, i.e. increase the overall signature generation time in favor of faster verification or vice-versa. We decrease the total number of iterations of f for some parameters, reducing the cost of key generation as well. Our main contribution allows one to choose a fixed cost with respect to the number of evaluations of f, unchanged for any input message. Our experiments show that these proposals substantially impact Merkle Tree based signature schemes, such as XMSS. Additionally, we give a formal proof that our scheme is Existentially Unforgeable under a Chosen Message Attack (EU-CMA), assuming that f is one way, second preimage resistant and undetectable function.

Keywords: Hash-based Signatures. Winternitz Signatures. Cryptography. Post-Quantum Cryptography. Combinatorics.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard.	41
COL	Collision Resistance.	38
EU-CMA	Existentially Unforgeable under a Chosen Message Attack.	17, 44
HBS	Hash Based Signatures.	31
LMS	Leighton-Micali Hash-Based Signatures.	31
MSS	Merkle Signature Scheme.	43
NAF	Non-Adjacent Form.	54
NIST	National Institute of Standards and Technology.	13, 31
OTS	One Time Signature.	31
OW	One-Wayness.	38
PKI	Public Key Infrastructure.	51
PQC	Post-Quantum Cryptography.	31
SHA-3	Secure Hash Algorithm 3.	41
SHA-2	Secure Hash Algorithm 2.	41
SPR	Second preimage resistance.	38
WOTS	Winternitz One Time Signature.	14, 31
WOTS+	Winternitz One Time Signature Plus.	16, 34
XMSS	eXtended Merkle Signature Scheme.	15, 31
XMSS^{MT}	Multi-Tree eXtended Merkle Signature Scheme.	51

CONTENTS

1 INTRODUCTION

Classical cryptography had reached its best with the development of rotor machines, such as the famous enigma machine recently portrayed in the Hollywood motion picture “The Imitation Game". Henceforth we have witnessed a shift of paradigm with the advent of personal computers; modern cryptography was born. The Data Encryption Standard (DES), developed in the early 70s, is a major milestone in the history of cryptography. However, and perhaps more importantly, in 1976 Public Key Cryptography (PKC) was first proposed by Whitfield Diffie and Martin Hellman in their paper entitled “New Directions in Cryptography" (DIFFIE; HELLMAN, 1976), which changed the way we use cryptography over the years.

Perhaps this is a moment of history recurrence, as we are at the brink of witnessing the creation of Quantum Computers. A computer with such power that threatens to break the majority of the cryptosystems used by billions of users over the internet. There are many unanswered questions in play: Will we ever have a practical quantum computer? If so, will it have computational capability comparable to theoretical expectations? Or maybe, for those in favor of conspiracy theories, what if a quantum computer already exists and we are not aware of it? Such questions have attracted the attention of the international community, calling out for solutions as we enter this new era of cryptography.

In this thesis, we study one of the oldest candidates of what we call today Post-Quantum Cryptography (PQC). That is, cryptography that can be deployed in ordinary computers that we have available today and considered safe against quantum adversaries. It is the case for Hash Based Signatures (HBS), schemes with well-known security reductions and considered safe against quantum computers. These schemes only use symmetric cryptography, more specifically cryptographic hash functions, which have been studied for decades. In addition, high security level hash function algorithms are believed to remain safe against quantum-computers. For this reason, there is little concern regarding the maturity and security of the current HBS state of the art. Nonetheless, it is not without its challenges, as HBS solves the signature problem in the quantum scenario by introducing its own limitations.

A cornerstone of HBS is the Winternitz One Time Signature (WOTS) scheme (MERKLE, 1989). As its name suggests, it has the capability of performing a single signature. However, distinct instances of the same One Time Signature (OTS) scheme can be grouped together into more complex schemes to achieve many-time signatures schemes; for example, the HBS many-time signature schemes EXtended Merkle Signature Scheme (XMSS) and Leighton-Micali Hash-Based Signatures (LMS), with public standards RFC 8391 (HÜLSING et al., 2018) and RFC 8554 (MCGREW; CURCIO; FLUHRER, 2019), respectively. Unfortunately, these solutions introduce a stateful condition, where the state of the private keys aggregated with a Merkle Tree has to be maintained safely. Hence, it followed from National Institute of Standards and Technology (NIST) that the use of stateful HBS standards is only to be considered for certain applications where the private key state integrity can be assured (COOPER et al., 2020). Finally, a more complex HBS scheme that removes the stateful constraint has been submitted to the ongoing NIST PQC standardization “competition[1]” (ALAGIC et al., 2020). This scheme is known as SPHINCS+ (BERNSTEIN et al., 2019; AUMASSON et al., 2020) and is CONSIDERED as alternative candidate in the third round of the submission process.

An interesting aspect of HBS is how the bit distribution of the message to be signed can impact the performance of some schemes. For example, the number of hash operations performed in the signature and verification steps of WOTS is directly determined by the input message. During these steps, the scheme uses an encoding algorithm that outputs a t-tuple of elements that uniquely represents the message. The elements of this tuple are then used to determine the number of hash operations in each step. This fact has been explored by (STEINWANDT; VILLÁNYI, 2008) and more recently by (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018) and (ROH; JUNG; KWON, 2018). These works propose different ways to encode a message such that either the signing process or the verification process can be performed using fewer hash functions. By doing so, other challenges arise. In general, we expect that by reducing the number of hash functions performed in the verification step entails that more hash functions have to be performed in the signature step, and vice versa. Indeed, it is the case that all the related work in the literature propose such trade-off in addition to an increased overall cost to the entire scheme. That is, key generation is also affected, and thus more hash operations are required to be performed in this step as well.

In this thesis, we raise the question: can we explore alternative encoding functions that do not increase key generation cost and allow such a trade-off at the same time? Are there encoding algorithms that may reduce the key generation cost, that is, reduce the entire cost of the scheme by decreasing the number of hash functions required for generating keys, signing and verifying signatures at the same time? Can we do all this efficiently? Finally, and perhaps the most important aspect of all, can we do this while preserving security, compared to the state of the art schemes in the literature?

OBJECTIVES

The Winternitz signature scheme utilizes a so called w parameter that introduces a clear trade-off between performance and signature length. However, the performance related to the number of hash functions required to produce or verify a Winternitz signature is also impacted by the message to be signed. The main goal of this thesis is to extend our knowledge about how do messages impact the performance of the Winternitz scheme and how we can use this to our advantage for different application scenarios. We evaluate different types of message encodings that can be used with the Winternitz scheme and how they affect it. As a result, we aim to develop and improve new techniques that may be used to lower the cost associated with the signature generation, the signature verification and the key generation of the WOTS based scheme. Our goal is to propose efficient algorithms with such properties, evaluate their

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

performance in practice and prove that the security of our scheme conforms to the latest state of the art.

MOTIVATION

Recently, NIST has published a call for candidates for post-quantum cryptography algorithm standards (NIST, 2016) . Furthermore, we have seen that chip manufactures have already achieved great advances in what could be the next step for the revolution of informatics, with Quantum Computers. Meanwhile, post-quantum cryptography is still in its early years, where security discussions and performance improvements are still constantly debated in public forums. While the international community is interested, it seems to be the right moment to contribute with a topic of such scale.

Considering the current state of PQC, the HBS candidates are possibly the most prominent alternatives when considering the immediate deployment of post-quantum signature schemes in the real world. While the maturity and security of these schemes are already well understood, we wish to push towards new alternatives that have not yet been thoroughly analyzed.

LIMITATIONS

There are plenty of alternatives to HBS schemes in the current state of the literature. Furthermore, it appears that a common use case for OTS schemes is their aggregation in Merkle Trees, to construct many-time signature schemes, under the same public key. Other interesting applications that are commonly referred to in the ongoing NIST standardization process are implementations for constrained devices, such as the Cortex-M4.

Therefore, we propose a narrowing of the scope, to be able to evaluate our contributions within this work. In this thesis, we address the encoding algorithms for WOTS based schemes. We consider its applications with experiments using XMSS reference code, which fits well to demonstrate the capability of our contributions, but do not provide results for other many-time signature schemes.

We address the performance of our algorithms and propose distinct techniques to reduce their running time. However, we do not discuss code optimizations, as we believe that our benchmark already demonstrates the limitations that we put forward for future works. Additionally, as we will see further on, there are numerous challenges to implement some of our contributions in embedded devices. Hence, while we do not discuss means to implement our algorithms in such platforms, we discuss how to get around these limitations for a specific set of application scenarios.

RESEARCH METHOD

For this thesis, we do not implement qualitative or quantitative research methods to gather and filter related works. We have found that very few works in the literature address the problem that we aim to investigate. Hence, we can easily summarize the state of the literature by addressing each work individually. The problem of integer encoding, on the other hand, is on a different level as it is a classical problem in Computer Science and Mathematics. However, we do not aim to survey the integer encoding problem in the scope of our work.

We will conduct our research by investigating the literature and enumerating distinct properties that are interesting for HBS. Mainly, we evaluate the trade-off gains of using different encoding algorithms with respect to the number of hash functions that are called in the signature scheme. Namely, we compare the average or the exact number of hash functions called for signing or verifying a message using distinct encoding functions. To provide a fair evaluation, we analyze these results considering similar or equal signature sizes for each alternative scheme, at the same security level.

Finally, we provide the necessary mathematical background to properly construct our proposals. To validate our work, we give estimates and/or exact results that can be used to compare with other methods. Furthermore, to address efficiency, we will conduct experiments that benchmark running time of our encoding alternative against previous works.

ACADEMIC CONTRIBUTIONS

In this thesis ,we explain our main contributions related to HBS, published in (PERIN et al., 2018) and (PERIN et al., 2021). The former is the result of a preliminary study of HBS. We analyze how messages affect the number of hash operations evaluated in the signature and verification steps of WOTS, then propose a technique to randomize this message, such that either signing or verifying can be performed with fewer hash iterations. The latter is the result of our main goal of this thesis. We found interesting properties in an encoding function called constant-sum as proposed by (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018). We updated the scheme so that its security level stands on the same ground as Winternitz One Time Signature Plus (WOTS+), for which we give a full and formal proof. In addition, we propose several improvements related to the encoding function, a deterministic algorithm and parameters that perform better overall, by decreasing the cost of key generation of our scheme.

Relevant mentions

During the development of this thesis, we have also engaged in other academic pursuits. Although the topic is not exactly aligned with the scope of this work, we include a summary of these contributions. We believe this to be relevant to convey our collaboration with distinguished and international researchers in multidisciplinary goals.

We are interested in understanding permutations generated by the evaluation of $g^i \pmod{p}$, for $i = 1, \dots p - 1$, p prime and where g is a primitive element of a multiplicative group of integers modulo p. Namely, g^i spans the set $\{1, \dots, p - 1\}$ and distinct choices of g for a fixed p produce distinct permutation sets.

This is an interesting problem as exponentiation of primitive elements of multiplicative groups of prime order plays a major role in several cryptographic primitives. This is the case, for example, in the ElGamal Signature scheme (ELGAMAL, 1985). We have studied and published evidence (NIEHUES et al., 2020) that these permutations preserve properties expected from random permutations.

Moreover, we continue this study with a different approach, producing v-ary sequences from these permutations. We call these sequences *ElGamal Sequences* (PANARIO; PERIN; STEVENS, N.D.). The sequence is quite simple, for some permutation $\pi = \{\pi_i : g^i \pmod{p}, 1 \leq i < p\}$, we denote an ElGamal sequence for some $v \mid p - 1$ as $\sigma = \{\sigma_i : \pi_i \pmod{v}\}$.

Example. *Let $g = 2$ and $p = 5$, then we have that $\pi = (2, 4, 1, 3)$. Since $2 \mid 5 - 1$, we have that $\sigma = (0, 0, 1, 1)$ for $v = 2$.*

In this paper, we give theoretical and experimental bounds of the number of runs and tuples of ElGamal Sequences in line with Golomb randomness postulates (GOLOMB; GONG, 2005). We also give bounds to sequences generated from random permutations and compare them to ElGamal Sequences.

ORGANIZATION

In Chapter 2, we give a brief introduction to basic concepts that are essential for most of our definitions and formal proofs. In Chapter 3, we explain the required background of HBS and OTS, describing WOTS+ scheme and the XMSS scheme. In Chapter 4, we review the alternative encoding algorithms that are used with WOTS in the literature. In Chapter 5 and Chapter 6, we present our main contributions as they are published in the literature. We keep the original work as published, with minor modifications to preserve the same notation throughout the thesis, and avoid repetition of background definitions. We conclude in Chapter 7 with final remarks, providing a discussion of the applicability of our work and our final thoughts considering both contributions.

2 PRELIMINARIES

2.1 FORMAL DEFINITIONS AND NOTATION

In this section, we introduce security concepts and notations that are required throughout this thesis. There is plenty of literature available for references, but we have opted to follow material that is publicly available and widely accepted by the international community, such as the work from Goldwasser & Bellare (2008) and Boneh & Shoup (2020).

2.1.1 General Notation

Definition 2.1.1 (Negligible). *A function v is negligible if for every constant $c \geq 0$, there exists an integer k_c such that $v(k) < k^{-c}$ for all $k \geq k_c$. For short we write "v is negligible in k" as* $\mathrm{negl}(k)$.

Definition 2.1.2 (Little-o). *For functions f, g if $0 \leq f(n) < c \cdot g(n)$ for all $c > 0$ and for all $n > k > 0$, then we say that $f(n) = o(g(n))$. The notation reads as "$f(n)$ is little-oh of $g(n)$" or "$f(n)$ is ultimately smaller than $g(n)$".*

Definition 2.1.3 (Advantage). *Given two distributions $\mathcal{X}$ and $\mathcal{Y}$, the advantage $\mathrm{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A})$ of an adversary $\mathcal{A}$ in distinguishing between these two distributions is defined as:*

$$\mathrm{Adv}_{\mathcal{X},\mathcal{Y}}(\mathcal{A}) = |\Pr[\mathcal{A}(\mathcal{X}) = 1] - \Pr[\mathcal{A}(\mathcal{Y}) = 1]|.$$

For simplicity, we consider that $\mathcal{A}$ is a probabilistic polynomial time algorithm. Informally, the advantage is a measurement of how well can $\mathcal{A}$ distinguish $\mathcal{X}$ from $\mathcal{Y}$, when given values of both distributions. It is commonly used in security proofs, where the goal is to show that $\mathcal{A}$ has negligible advantage distinguishing two distributions.

Definition 2.1.4 (Statistical distance). *Suppose $\mathcal{X}$ and $\mathcal{Y}$ are probability distributions on a finite set $\mathcal{R}$. Then their* statistical distance *is defined as*

$$\Delta(\mathcal{X},\mathcal{Y}) = \frac{1}{2}\sum_{\alpha\in\mathcal{R}} |\Pr[\alpha = \mathcal{X}] - \Pr[\alpha = \mathcal{Y}]|.$$

We use the statistical distance of two distributions as another measurement of similarity of $\mathcal{X}$ and $\mathcal{Y}$. We are interested in showing that for some set $\mathcal{R}$ of size k then $\Delta(\mathcal{X},\mathcal{Y}) = \mathrm{negl}(k)$. That is, the distributions $\mathcal{X}$ and $\mathcal{Y}$ are *statistically indistinguishable*.

We use the notation $u \leftarrow\!\$ \{0,1\}^{\ell}$ to denote sampling a ℓ-bit string uniformly at random. We also use the notation $\mathcal{U}_\lambda$ to denote the uniform distribution over λ-bit strings or the notation $u \sim \mathcal{U}_\lambda$ to denote that u follows the $\mathcal{U}_\lambda$ distribution. The repeated evaluation of a function f on some input x is recursively defined as $f^0(x) = x$ and $f^i(x) = f(f^{i-1}(x))$ for any non-negative integer i. In Table 1, we summarize some notations that we frequently use throughout the

Table 1 – Common notation

M	Input message to be signed
ℓ	Length of the input message (in number of bits)
$\mathcal{M}$	A message space
λ	Security parameter
$\leftarrow\$$	Sampling uniformly at random
$\overset{?}{=}$	Compare and return if both sides are equal
$\|\|x\|\|$	Denotes the norm of x
$x\|\|y$	The concatenation of two strings x and y
$\mathcal{H}$	A cryptographic hash function.
$\mathcal{H}_\ell$	A cryptographic hash function with output length ℓ.
$\|\mathcal{S}\|$	The cardinality of a set of integers $\mathcal{S}$
$\min(\mathcal{S})$	The minimum element of a set of integers $\mathcal{S}$
$\max(\mathcal{S})$	The maximum element of a set of integers $\mathcal{S}$

Source: The author.

thesis. Namely, unless specified otherwise, the following notation should always serve the same purpose.

Let $M = (m_1, \ldots, m_\ell)$ be a binary string of length ℓ. We denote $M_{a:b}$, for a, b non-negative integers where $a \leq b$, as the sub-string $(m_a, \ldots, m_b)$. In the particular case where $b = \ell$, we omit b by simply stating $M_{a:}$. In the case $a > b$, then $M_{a:b}$ is an empty string. Lastly, we use the vector notation such as $\mathbf{r} = (r_1, \ldots, r_\ell)$ with bold fonts to properly distinguish the vector from its elements.

2.1.2 Hash function families

In this section, we closely follow (HÜLSING, 2013a; HÜLSING, 2013b; ROGAWAY; SHRIMPTON, 2004) due to our security proof being strongly aligned with these works.

Let $\mathcal{F}_\lambda : \{f_k : \mathcal{M} \to \{0,1\}^\lambda \mid k \in \mathcal{K}_\lambda\}$ be a family of functions with message space $\mathcal{M} = \{0,1\}^\ell$ and key space $\mathcal{K}_\lambda = \{0,1\}^\lambda$. For simplicity, we always consider that sampling $k \leftarrow\$ \mathcal{K}_\lambda$ and evaluating functions from $\mathcal{F}_\lambda$ can be done efficiently in polynomial time. Moreover, if $\ell > \lambda$ we call $\mathcal{F}_\lambda$ a hash function family. Then we can define the success probability of an adversary $\mathcal{A}$ against the One-Wayness (OW) of $\mathcal{F}_\lambda$ as:

$$\mathrm{Succ}^{\mathrm{OW}}_{\mathcal{F}_\lambda}(\mathcal{A}) = \Pr[k \leftarrow\$ \mathcal{K}_\lambda,\ x \leftarrow\$ \mathcal{M},\ y \leftarrow f_k(x),\ x' \leftarrow\$ \mathcal{A}(k,y) : y = f_k(x')],$$

against the Second preimage resistance (SPR) of $\mathcal{F}_\lambda$ as:

$$\mathrm{Succ}^{\mathrm{SPR}}_{\mathcal{F}_\lambda}(\mathcal{A}) = \Pr[k \leftarrow\$ \mathcal{K}_\lambda,\ x \leftarrow\$ \mathcal{M},\ x' \leftarrow \mathcal{A}(k,x) : (x \neq x') \wedge (f_k(x) = f_k(x'))],$$

and against the Collision Resistance (COL) of $\mathcal{F}_\lambda$ as:

$$\mathrm{Succ}^{\mathrm{COL}}_{\mathcal{F}_\lambda}(\mathcal{A}) = \Pr[k \leftarrow\$ \mathcal{K}_\lambda,\ (x,x') \leftarrow \mathcal{A}(k) : (x \neq x') \wedge (f_k(x) = f_k(x'))].$$

Now let InSec denote the maximum success probability of all possible probabilistic adversaries $\mathcal{A}$ running in time less than or equal to z. We define one-wayness, second preimage resistance and collision resistance as follows.

Definition 2.1.5 (OW). *Let λ be the security parameter, then $\mathcal{F}_\lambda$ is one-way if*

$$\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda; z) = \max_{\mathcal{A}}(\mathrm{Succ}^{\mathrm{OW}}_{\mathcal{F}_\lambda}(\mathcal{A})) = \mathrm{negl}(\lambda).$$

Definition 2.1.6 (SPR). *Let λ be the security parameter, then $\mathcal{F}_\lambda$ is second preimage resistant if*

$$\mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda; z) = \max_{\mathcal{A}}(\mathrm{Succ}^{\mathrm{SPR}}_{\mathcal{F}_\lambda}(\mathcal{A})) = \mathrm{negl}(\lambda).$$

Definition 2.1.7 (COL). *Let λ be the security parameter, then $\mathcal{F}_\lambda$ is collision resistant if*

$$\mathrm{InSec}^{\mathrm{COL}}(\mathcal{F}_\lambda; z) = \max_{\mathcal{A}}(\mathrm{Succ}^{\mathrm{COL}}_{\mathcal{F}_\lambda}(\mathcal{A})) = \mathrm{negl}(\lambda).$$

In short, we assume that $\mathcal{A}$ always has access to f_k. In the case of OW, the adversary obtains y and outputs a message x' such that $y = f_k(x')$. This message x' may be equal to the original value used to obtain y or a different one, either case break OW. In the case that $\mathcal{A}$ has access to the original message x, then SPR is related to the hardness of finding $x' \neq x$ such that $f_k(x') = y$. Lastly, for COL, the adversary has the capability of choosing both x and x' such that $f_k(x) = f_k(x')$ and $x \neq x'$. We say that COL is a stronger security assumption for the function family $\mathcal{F}_\lambda$ due to the birthday paradox. Namely, it is usually easier for $\mathcal{A}$ to break COL than it is for OW or SPR. We remark that COL implies SPR which consequently implies OW, however, the reciprocal does not hold (BONEH; SHOUP, 2020, Definition 8.6).

Now we define the undetectability (UD) property of $\mathcal{F}_\lambda$. For distributions over $\{0,1\}^\lambda \times \mathcal{K}_\lambda$, we let a sample (u, k) from distribution $\mathcal{D}_{\mathrm{UD},\mathcal{U}}$ be obtained by choosing $u \leftarrow\$ \mathcal{U}_\lambda$ and $k \leftarrow\$ \mathcal{K}_\lambda$ uniformly at random. Otherwise, we let a sample (u, k) be obtained from distribution $\mathcal{D}_{\mathrm{UD},\mathcal{F}}$ by letting u as the evaluation of f_k on a uniformly distributed λ-bit string, where $k \leftarrow\$ \mathcal{K}_\lambda$. In other words, by allowing a slight abuse of notation, we write $u \leftarrow f_k(\mathcal{U}_\lambda)$. From this, we define the distinguishing advantage of an adversary, followed by the undetectability of $\mathcal{F}_\lambda$.

Definition 2.1.8 (UD). *Let λ be the security parameter, then $\mathcal{F}_\lambda$ is undetectable if*

$$\mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_\lambda; z) = \max_{\mathcal{A}}(\mathrm{Adv}_{\mathcal{D}_{\mathrm{UD},\mathcal{U}},\mathcal{D}_{\mathrm{UD},\mathcal{F}}}(\mathcal{A})) = \mathrm{negl}(\lambda).$$

The existence of function families with such properties is still an open problem. However, it is known that SPR hash function families can be obtained with OW functions, and that OW implies the existence of secure signature schemes (ROMPEL, 1990). A similar result exists for UD showing that the existence of OW is equivalent to the existence of pseudorandom bit generator (HÅSTAD et al., 1999). Therefore, following the rationale in (HÜLSING, 2013a), it seems reasonable to assume that one can achieve SPR, OW and UD. Similar implications are not known for COL hash function families, so their existence is a stronger assumption.

2.1.3 Digital Signatures Schemes

In one of the contributions of our work, we describe and propose a one-time signature scheme and prove its security. Hence, we aim to give the necessary definitions that we follow for better description and understanding of the content. First, we give the classic definition of *signature schemes*.

Definition 2.1.9 ((HÜLSING, 2013b, p. 3)). *Let $\mathcal{M}$ be the message space. A digital signature scheme $\mathcal{S} = (\textsc{Gen}, \textsc{Sig}, \textsc{Ver})$ is a triple of probabilistic polynomial time algorithms:*

Gen (1^λ). *On input of a security parameter 1^λ outputs a secret key* **sk** *and public key* **pk***;*

Sig (M, sk). *Outputs a signature σ under* **sk** *for the message $M \in \mathcal{M}$;*

Ver (M, σ, pk). *Outputs 1 if and only if σ is a valid signature on M under* **pk***;*

such that

$$\forall(\mathbf{pk}, \mathbf{sk}) \leftarrow \textsc{Gen}(1^\lambda), \forall(M \in \mathcal{M}) : \textsc{Ver}(\mathbf{pk}, \textsc{Sig}(\mathbf{sk}, M), M) = 1.$$

Now we introduce the definition by Boneh et al. (BONEH; SHEN; WATERS, 2006) of a digital signature scheme being existentially unforgeable under an adaptive chosen-message attack (EU-CMA). This is based on a game with three phases, given below.

Setup. The challenger runs $\textsc{Gen}(1^\lambda)$, obtains $(\mathbf{sk}, \mathbf{pk})$, provides the adversary $\mathcal{A}$ with **pk** and keeps **sk** to itself.

Query. The adversary $\mathcal{A}$ submits signature queries of messages $M_1, \ldots, M_q$ of its choice to the challenger, who replies with $\sigma_1 = \textsc{Sig}(M_1, \mathbf{sk}), \ldots, \sigma_q = \textsc{Sig}(M_q, \mathbf{sk})$. These queries can be made adaptively by $\mathcal{A}$.

Output. The adversary outputs a pair (M', σ') and wins the game if $\textsc{Ver}(M', \sigma', \mathbf{pk})$ is true and $M' \notin \{M_1, \ldots, M_q\}$.

We observe that $\mathcal{A}$ is always considered to be a probabilistic polynomial time algorithm. The formal definition of EU-CMA is given below.

Definition 2.1.10 (EU-CMA (BONEH; SHEN; WATERS, 2006, Def. 1)). *A signature scheme is (z, q, ϵ)*-existentially unforgeable under an adaptive chosen-message attack *if no z-time adversary $\mathcal{A}$ making at most q signature queries has advantage at least ϵ in the above game.*

Remark. *In Section 6.5.2 we give a formal proof of our scheme using Definition 2.1.10. Since the main object of this thesis are one-time signature schemes, we use $q = 1$.*

2.2 SECURITY LEVELS AND ATTACK BOUNDS

A common approach to evaluate security levels of signature schemes is the "λ-bit security" term, compatible with the security evaluation of symmetric cryptosystems such as the Advanced Encryption Standard (AES). For example, this has been a requirement for all submissions to the ongoing PQC standardization process held by NIST, to evaluate classical and quantum security (NIST, 2016). In this section, we describe the security level of $\mathcal{F}_\lambda$ with respect to OW, SPR and UD properties.

According to Lenstra (2004), we say that a symmetric cryptosystem with λ-bit keys has λ-bit security if it does not allow a generic attack to be faster than an exhaustive key search of 2^λ keys — or $2^{\lambda-1}$ in the average case. Then, by following (HÜLSING, 2013b) and (DODS; SMART; STAM, 2005), we can assume for classical generic attacks

$$\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_\lambda; z) = \frac{z}{2^\lambda}.$$

Regarding the recent scenario and motivation of our work, we also give bounds for quantum attacks against $\mathcal{F}_\lambda$. For a conservative approach, we consider that the security levels of symmetric cryptosystems are reduced from λ bits to $\lambda/2$ bits, due to Grover (1996). However, as it turns out, it seems that Grover's quantum attack fails to achieve theory in practice. For example, the latest security bounds for AES with 256-bit classical security places it at approximately 163-bit security level against quantum adversaries (GHEORGHIU; MOSCA, 2021). For cryptographic hash functions, the same research claims that a lower bound for the security level of Secure Hash Algorithm 2 (SHA-2) and Secure Hash Algorithm 3 (SHA-3) of 256-bit output length stands at approximately 165 bits. The point of interest in such a claim is that HBS is commonly instantiated with these hash functions. Nonetheless, for the sake of comparability, we will use the conservative approach and assume that

$$\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_\lambda; z) = \frac{z}{2^{\frac{\lambda}{2}}}$$

versus quantum adversaries. With this, we hope to avoid any misinterpretation claiming that we present schemes with higher security levels than those available in the literature.

3 HASH BASED SIGNATURES

3.1 ONE-TIME SIGNATURE SCHEMES

As the main contribution of this thesis is a novel hash-based OTS, we give a brief introduction to HBS in this Chapter. The main reason why HBS has become so important recently, is due to the common belief that such schemes can be deployed safely, in a scenario where a quantum computer attack against current cryptographic protocols becomes practical. As we have seen previously, symmetric cryptosystems are expected to remain secure, even in the aftermath of a full fledged quantum computer. However, as this possibility only gained momentum throughout the last decade, an early adoption of such schemes was not a common concern among cryptographers and the international communities.

The schemes detailed in this chapter contain several limitations that are still being worked and improved in the literature. To name a few, most HBS schemes can usually be used one time and need to be combined by using Merkle Trees to allow additional signatures to be verified against a common public key. Furthermore, the introduction of a Merkle Tree to aggregate OTS public keys introduces a new challenge, which is that of the statefulness of the resulting set of secret keys. In other words, the security of the scheme also relies on mechanisms that can guarantee that the state of the key is preserved: no two distinct signatures may be published under the same secret key.

We will not tackle the problem of stateful and stateless schemes in this thesis. Our goal lies within the encoding algorithms that are used to sign messages. In the following, we explain the Lamport OTS as an introductory contextualization to the more advanced schemes. Following up, we give Wots+, which will be the building block for our main contribution. We finalize by giving a brief introduction to Merkle Signature Scheme (MSS) and variants, as it will be useful to understand the main case of application scenarios and the performance experiments we conduct further on.

3.1.1 Lamport one-time signature scheme

In 1979, Leslie Lamport designed a signature scheme that today is widely known as Lamport Signatures (LAMPORT, 1979). This scheme is the fundamental construction used by many modern HBS schemes. To define the scheme, we start by letting $f = f_k \in \mathcal{F}_\lambda$ denote any one-way function as defined in the previous section. Let M be a message of length ℓ, namely $M \in \{0,1\}^\ell$, then the scheme is defined as follows:

Gen (1^λ). Pick at random 2ℓ strings of λ bits to create the secret key

$$\mathbf{sk} = (\mathrm{sk}_{1,0}, \mathrm{sk}_{1,1}, \ldots, \mathrm{sk}_{\ell,0}, \mathrm{sk}_{\ell,1}).$$

The public key is obtained by applying f to each element of the secret key:

$$\mathbf{pk} = (\mathrm{pk}_{1,0}, \mathrm{pk}_{1,1}, \ldots, \mathrm{pk}_{\ell,0}, \mathrm{pk}_{\ell,0}) = (f(\mathrm{sk}_{1,0}), f(\mathrm{sk}_{1,1}), \ldots, f(\mathrm{sk}_{\ell,0}), f(\mathrm{sk}_{\ell,0})).$$

Sig $(M, \mathbf{sk})$. A signature for a message $M = (m_1, \ldots m_\ell) \in \{0,1\}^\ell$ consists making public the elements of the private key, corresponding to the position and binary value of the input message:

$$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_\ell) = (\mathrm{sk}_{1,m_1}, \ldots \mathrm{sk}_{\ell,m_\ell}).$$

Ver $(M, \boldsymbol{\sigma}, \mathbf{pk})$. A signature $\boldsymbol{\sigma}$ is verified by comparing the evaluation using f of each element of the signature with the corresponding element of the public key:

$$(f(\sigma_1), \ldots, f(\sigma_\ell) \overset{?}{=} (\mathrm{pk}_{1,m_1}, \ldots, \mathrm{pk}_{\ell,m_\ell}).$$

This short description gives an idea of how the Lamport OTS scheme is a quite simple construction. Indeed, the scheme can be proven Existentially Unforgeable under a Chosen Message Attack (EU-CMA) solely on the assumption that f is OW (BERNSTEIN; BUCHMANN; DAHMEN, 2008).

3.1.2 Winternitz one-time signature scheme

The Winternitz One-Time Signature scheme, named hereon by its popular acronym Wots, is a specialization of Lamport Signatures (LAMPORT, 1979). We can see the latter as signing individual bits of a binary input string, whereas the former signs multiple bits at a time. As we will see in the coming definitions, a clear advantage of this is the reduced signature length. Moreover, to avoid repetition, we will only give the definition of the state of the art variant Wots+. The main difference of this scheme is the introduction of a random vector $\mathbf{r}$ in the chaining function, which reduces the original security assumption on f_k from collision resistance to second preimage resistance (HÜLSING, 2013b; BUCHMANN et al., 2011).

First, consider the following definitions that we use throughout the paper. We denote w as the *Winternitz parameter*, where w and ℓ are positive non-zero integers with $1 \leq w \leq 2^\ell$. Let λ be the security parameter and message space $\mathcal{M} = \{0,1\}^\ell$, then we have

$$t_1 = \left\lceil \frac{\ell}{\log_2 w} \right\rceil, \quad t_2 = \left\lfloor \frac{\log_2 t_1(w-1)}{\log_2 w} \right\rfloor + 1$$

$$\text{and } t = t_1 + t_2.$$

Definition 3.1.1. *Let $c_k^i(x, \mathbf{r})$ be denoted as the Winternitz* chaining *function, where $i \in \mathbb{N}$, $k \in \mathcal{K}_\lambda$, $f_k \in \mathcal{F}_\lambda$, $x \in \{0,1\}^\lambda$ and $\mathbf{r} = (r_1, \ldots, r_j) \in \{0,1\}^{\lambda \times j}$ with $j \geq i$. This function is defined recursively as*

$$c_k^i(x, \mathbf{r}) = \begin{cases} x & \text{if } i = 0, \\ f_k(c_k^{i-1}(x, \mathbf{r}) \oplus r_i) & \text{otherwise.} \end{cases}$$

Remark. *As a common way of instantiating the Winternitz chaining function is by letting f_k be a cryptographic hash function (such as SHA-2 or SHA-3), we sometimes refer to the Winternitz chaining function as hash chains.*

With these definitions, we can now define WOTS+ scheme:

GEN (1^λ). Sample t strings of λ bits uniformly at random to compose the secret key

$$\mathbf{sk} = (\mathrm{sk}_1, \ldots, \mathrm{sk}_t) = (x_1, \ldots, x_t).$$

By choosing $k \leftarrow\$ \mathcal{K}_\lambda$ and $\mathbf{r} = (r_1, \ldots, r_{w-1}) \leftarrow\$ \{0,1\}^{\lambda \times w-1}$, compute the public key

$$\mathbf{pk} = (\mathrm{pk}_0, \mathrm{pk}_1, \ldots, \mathrm{pk}_t) = \big((\mathbf{r}, k), c_k^{w-1}(x_1, \mathbf{r}), \ldots, c_k^{w-1}(x_t, \mathbf{r})\big).$$

SIG (M, sk). Take an ℓ-bit message $M \in \mathcal{M}$ and represent it as a t_1-tuple of base-w words, i.e., $\mathcal{B}_1 = (b_1, \ldots, b_{t_1})$. The generic version of this encoding is referred to as `base-w` in RFC 8391 (HÜLSING et al., 2018) and described in Section 3.1.3. A checksum is computed by considering the integer representation of each element in $\mathcal{B}_1$ as $Q = \sum_{j=1}^{t_1}(w - 1 - b_j)$. Finally, applying `base-w` to Q analogously yields $\mathcal{B}_2 = (b_{t_1+1}, \ldots, b_t)$. Let $\mathcal{B} = (b_1, \ldots, b_t)$, the concatenated `base-w` representation of M and Q. The signature is generated as

$$\begin{aligned}\boldsymbol{\sigma} &= (\sigma_1, \ldots, \sigma_t)\\ &= (c_k^{b_1}(x_1, \mathbf{r}), \ldots, c_k^{b_t}(x_t, \mathbf{r})).\end{aligned}$$

VER ($M, \boldsymbol{\sigma}$, pk). Obtain the t-tuple $\mathcal{B}$ as described in SIG. The correctness of $\boldsymbol{\sigma}$ is asserted by evaluating the remaining iterations over signature blocks and comparing the results with the public key:

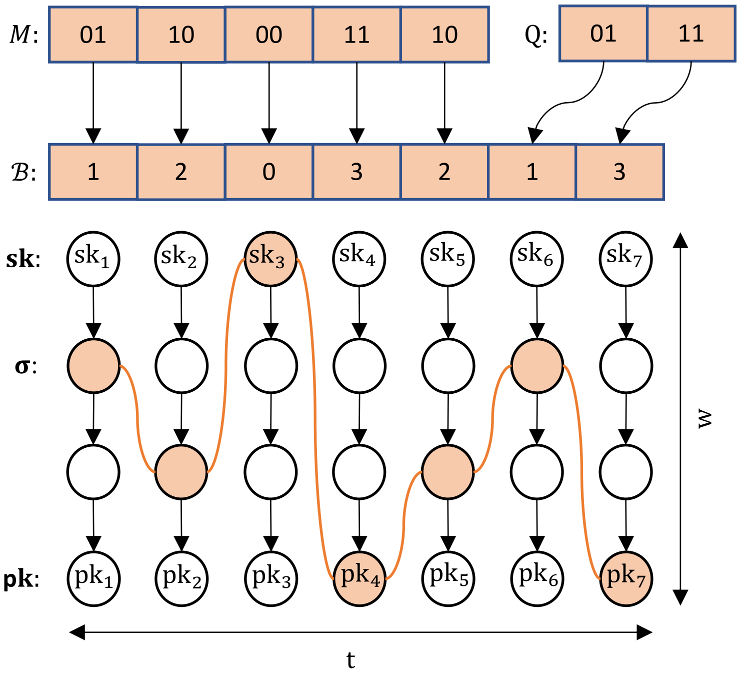
$$\mathbf{pk} \stackrel{?}{=} ((\mathbf{r}, k), c_k^{w-1-b_1}(\sigma_1, \mathbf{r}_{b_1+1:}), \ldots, c_k^{w-1-b_t}(\sigma_t, \mathbf{r}_{b_t+1:})).$$

Figure 1 illustrates WOTS+ and how the keys are obtained through the hash chains. We also give a short example to demonstrate how the indexing of $\mathbf{r}$ is used to produce the correct public key from each corresponding signature element.

Example. *Let $w = 8$ and consider the signature and verification of a single position i where $b_i = 3$ from $\mathcal{B}$, for the sake of the example. Then sign with the $\mathrm{sk}_i = x$ by evaluating $\sigma_i = c_k^3(x, \mathbf{r})$ and verify the signature by asserting that $\mathrm{pk}_i \stackrel{?}{=} c_k^{8-1-3}(\sigma_i, \mathbf{r}_{3+1:}) = c_k^4(\sigma_i, \mathbf{r}_{4:})$. Observe that the notation of the sub-vector of $\mathbf{r}$ guarantees that c_k starts at the correct position of $\mathbf{r}$, completing the entire hash chain, and thus ensuring the signature verification.*

$$\mathbf{r} = (\underbrace{r_1, r_2, r_3}_{\textit{SIG}}, \underbrace{r_4, r_5, r_6, r_7}_{\textit{VER}}).$$

Figure 1 – Example of WOTS+ with $\ell = 10$ and $w = 4$.



Source: The author.

There are significant improvements achieved by WOTS+ over Lamport signatures. The first one is that the public key can be compacted into a short digest, using what is referred to in the literature as an L-tree. In summary, it consists of a tree-like structure that hashes pairs of public key elements with bitmasks, producing a short unique output. The verification of the signature can be performed comparing the root of the L-tree obtained from the public key obtained in VER. One can verify that this is not possible in the case of Lamport signatures, as we need all elements of the public key to be able to compare to the signature. Another interesting characteristic is the ability to significantly decrease the size of the signature $\|\boldsymbol{\sigma}\| = t\lambda$ by selecting larger w. This comes as a trade-off choice, since the number of iterations of the Winternitz chaining function will increase with w. In Table 2, we show these values where $\|\boldsymbol{\sigma}\|$ is the size of the signature in bytes and $C(\text{GEN})$ is the total cost of generating the public key with $t(w-1)$ evaluations of f_k. We discuss costs related to SIG and VER in Section 5.4.

3.1.3 Domination Free Functions and the `base-w` encoding

In this thesis, we investigate encoding functions with certain properties. We find that such functions are clearly defined in (BONEH; SHOUP, 2020) generically to be used with HBS. Let $\mathcal{I}_{(t,n)} = \{(b_1, \ldots, b_t) : 0 \leq b_i \leq n\}$ denote the set of t-tuples of positive integers not larger than n. We denote $\mathcal{B} = (b_1, \ldots, b_t) \in \mathcal{I}_{(t,n)}$ with $0 \leq b_i \leq n$ for $i = 1, \ldots, t$. Then we introduce the following definition.

Table 2 – Parameters of WOTS+ for $\ell = \lambda \in \{256, 512\}$.

ℓ	t	w	$\|\sigma\|$ in bytes	$C(\textsc{Gen})$
256	67	16	2144	1005
	45	64	1440	2835
	34	256	1088	8670
512	131	16	8384	1965
	89	64	5696	5607
	66	256	4224	16830

Source: The author.

Definition 3.1.2 ((BONEH; SHOUP, 2020, Def. 14.4)). *Let $\mathcal{B}, \mathcal{B}' \in \mathcal{I}_{(t,n)}$, then $\mathcal{B}'$ dominates $\mathcal{B}$ if $b'_i \geq b_i$ for all $i = 1, \ldots, n$. Moreover, we say that a function $P : \mathcal{M} \to \mathcal{I}_{(t,n)}$ is* domination free *if for all distinct messages $M, M' \in \mathcal{M}$ the vector $P(M')$ does not dominate $P(M)$.*

Let $n = w - 1$ and $\mathbf{E} : \{0,1\}^\ell \to \mathcal{I}_{(t,n)}$ denote the `base-w` encoding of M and Q defined in the previous section. To help visualize the properties of the `base-w` encoding, we first give an example with small parameters containing all evaluations of $\mathbf{E}(M)$ with $M \in \{0,1\}^\ell$ in Table 3. We observe that the checksum $\mathcal{B}_2$ guarantees that if any element of $\mathcal{B}_1$ is increased, then $\mathcal{B}_1$ is decreased. With this, we give a short proof showing that $\mathbf{E}$ is *domination free*.

Table 3 – `base-w` encoding example for $\ell = 4$ and $w = 4$, we get $t_1 = 2$ and $t_2 = 2$,

M	$\mathbf{E}(M) = (b_1, b_2, b_3, b_4)$	M	$\mathbf{E}(M) = (b_1, b_2, b_3, b_4)$
0000	(0, 0, 1, 2)	1000	(2, 0, 1, 0)
0001	(0, 1, 1, 1)	1001	(2, 1, 0, 3)
0010	(0, 2, 1, 0)	1010	(2, 2, 0, 2)
0011	(0, 3, 0, 3)	1011	(2, 3, 0, 1)
0100	(1, 0, 1, 1)	1100	(3, 0, 0, 3)
0101	(1, 1, 1, 0)	1101	(3, 1, 0, 2)
0110	(1, 2, 0, 3)	1110	(3, 2, 0, 1)
0111	(1, 3, 0, 2)	1111	(3, 3, 0, 0)

Source: The author.

Lemma 3.1.1 ((BONEH; SHOUP, 2020, Lem. 14.5)). *For every distinct $M, M' \in \{0,1\}^\ell$ we have that $\mathbf{E}(M)$ does not dominate $\mathbf{E}(M')$.*

Proof. Let $\mathcal{B} = \mathbf{E}(M)$, $\mathcal{B}' = \mathbf{E}(M')$ and assume $\mathcal{B}'$ dominates $\mathcal{B}$. Because $\mathbf{E}$ is injective, it follows that $\mathcal{B} \neq \mathcal{B}'$. Therefore, there exists an i where $1 \leq i \leq t$ such that $b'_i > b_i$. If $i \leq t_1$, we must have that $\mathcal{B}'_2$ does not dominate $\mathcal{B}_2$. Otherwise, if $i > t_1$, we must have that $\mathcal{B}'_1$ does not dominate $\mathcal{B}_1$. □

Remark. *The previous lemma and proof have been slightly modified from the original version, to incorporate our distinct notation, but following the same rationale.*

3.1.4 Estimated costs of Winternitz signatures

Let the number of iterations of f_k for $\textsc{Gen}$ be denoted as $C(\textsc{Gen})$ and analogously for $\textsc{Sig}$ and $\textsc{Ver}$. Evidently, $C(\textsc{Gen}) = C(\textsc{Sig}) + C(\textsc{Ver})$. In the case of $\textsc{Wots+}$, we have that $C(\textsc{Gen}) = t(w-1)$. For now, we assume that all elements of $\mathcal{B}$ are uniformly distributed at random, under the same assumption for M; that is, $M \sim \mathcal{U}_\ell$. This allows us to estimate costs, on average, of $\overline{C(\textsc{Sig})} \approx \overline{C(\textsc{Ver})} \approx \frac{1}{2}C(\textsc{Gen})$. Hence parameters for $\textsc{Wots+}$ from (HÜLSING et al., 2018) are presented in Table 4, associated with their respective average costs. Moreover, we give more information on the distribution of $\mathcal{B}_1$ and $\mathcal{B}_2$ in Chapter 5, providing evidence to how these tuples behave in practice. We also give a more elaborate discussion in Section 5.4.

Table 4 – Number of iterations of c_k for usual parameters of $\textsc{Wots+}$.

m	t	w	$C(\textsc{Gen})$	$\approx \overline{C(\textsc{Sig})} \mid \overline{C(\textsc{Ver})}$
256	67	16	1005	502.5
	45	64	2835	1417.5
	34	256	8670	4335.0
512	131	16	1965	982.5
	89	64	5607	2803.5
	66	256	16830	8415.0

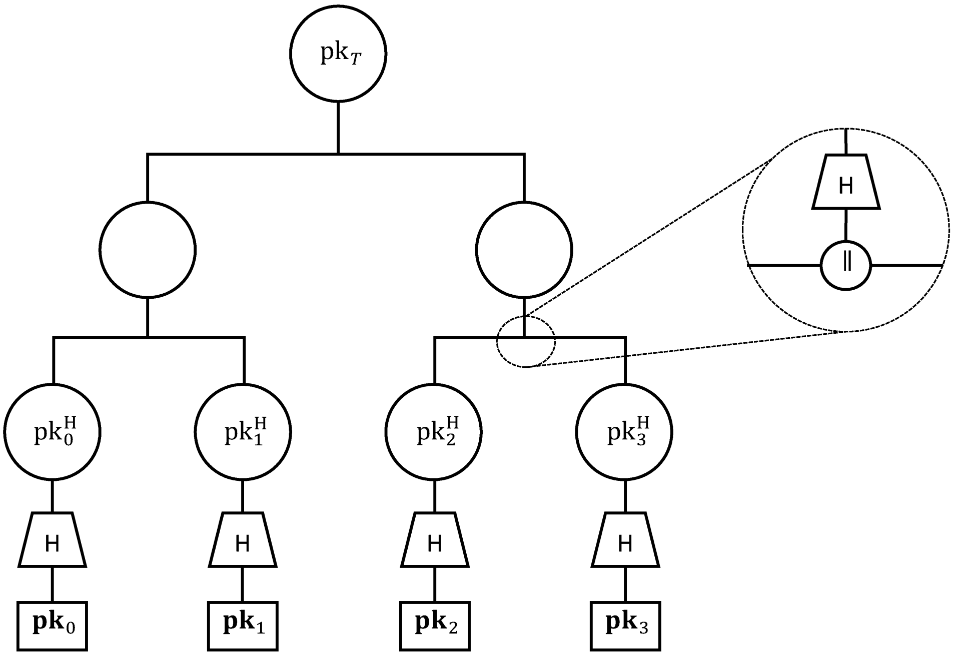
Source: The author.

Henceforth, we use these estimates to be able to compare the cost associated with the `base-w` encoding to other encoding alternatives. For this, we always assume that M is uniform and random and costs are taken on average, unless stated otherwise. Then, for simplicity, we proceed by considering the interpretation of (signature, verification or generation) “costs” as the definitions given in this subsection.

3.2 MERKLE TREE SIGNATURE SCHEMES

A clear limitation of OTS, in general, is that only one signature may be performed under the same key pair. To avoid this issue, it was proposed by Ralph Merkle in (MERKLE, 1989) that many OTS public keys could be aggregated into one single and short public key. This was a revolutionary step towards many-time HBS schemes, solving the problem with distributing numerous public keys efficiently. The goal of the tree structure is to set all leaf nodes as the hash of the public keys of multiple instances of an OTS scheme. Then proceed by hashing concatenated pairs of child nodes, recursively, until a single node is obtained. Namely, the root node. Now, instead of returning public keys of the many underlying OTS, the scheme returns the root of the tree as a unique public key to all OTS that generated the leaves; See Figure 2. This signature scheme is called MSS, and the aggregation with the tree structure is often referred to as Merkle Trees.

Figure 2 – Example of MSS with $h = 2$.


pk_T
H
||
$\mathrm{pk}^{\mathrm{H}}_0$
$\mathrm{pk}^{\mathrm{H}}_1$
$\mathrm{pk}^{\mathrm{H}}_2$
$\mathrm{pk}^{\mathrm{H}}_3$
H
H
H
H
$\mathbf{pk}_0$
$\mathbf{pk}_1$
$\mathbf{pk}_2$
$\mathbf{pk}_3$

Source: The author.

Let h be the height of the Merkle Tree, then for some OTS, a brief definition of MSS is as follows:

Gen (1^λ)**.** Generate 2^h OTS key pairs and compute $(\mathbf{sk}_T, \mathrm{pk}_T)$ as

$$\mathbf{sk}_T = (\mathbf{sk}_0, \ldots, \mathbf{sk}_{2^h-1}) \text{ and}$$

$$\mathrm{pk}_T = \mathrm{Root}(\mathrm{pk}^H_0, \ldots, \mathrm{pk}^H_{2^h-1}),$$

where pk^H_i is the hash of the OTS in position i and Root is a function that computes the root of the tree with all the OTS hashed public keys.

Sig $(M, i, \mathbf{sk}_T)$**.** A signature for a message $M \in \mathcal{M}$ is generated by computing the OTS signature σ_i of the i-th leaf node. Then, compose the signature as

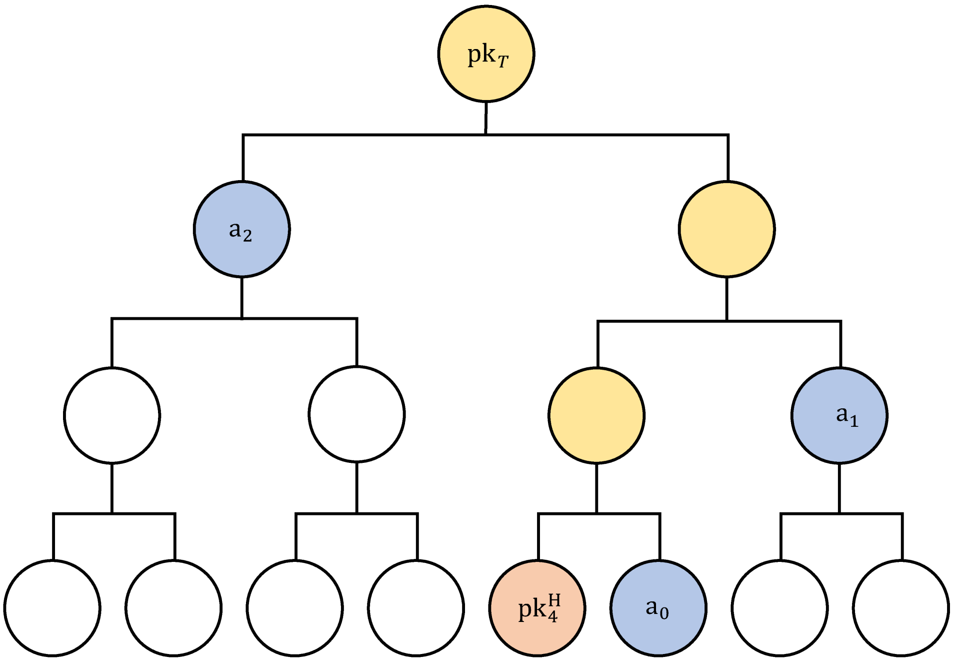
$$\sigma_T = (i, \sigma_i, \mathbf{pk}_i, \mathbf{Auth}_i)$$

where $\mathbf{Auth}_i$ is the ordered set with all sibling nodes required to compute pk_T from $\mathbf{pk}_i$.

Ver $(M, \sigma_T, \mathrm{pk}_T)$**.** The verifier initially verifies the signature with the OTS using M, σ_i and $\mathbf{pk}_i$ from σ_T. If valid, then it verifies the membership of the obtained public key in the tree structure, by computing its root node with $\mathbf{Auth}_i$ provided in σ_T. That is, the tree public key pk_T.

Remark. *The scheme definition does not contain any mechanism to control repeated uses of the same key. This shows the limitation of MSS and variations, where the state of the private key has to be carefully maintained. In other words* $\textsc{Sig}\,(M, i, \mathbf{sk}_T)$ *can only be called once for the same* i*.*

Figure 3 – Example of MSS authentication path $\mathrm{Auth}_4 = (a_0, a_1, a_2)$ for $h = 3$.



Source: The author.

The choice of h for MSS determines the number of signatures that the scheme can perform. It also has an impact on the performance of the signature step, when considering that $\mathbf{Auth}_i$ has to be uniquely computed for each i. It turns out that the computation of $\mathbf{Auth}_i$ may be the most costly operation in the signature step, for example, in the case that the public keys pk_i^H are not stored in memory. In this case, to compute $\mathbf{Auth}_i$, the signer would be required to nearly recompute the entire tree structure. On the other hand, storing 2^h nodes may be quite expensive and inconvenient for memory constrained platforms. One possibility to reduce this overhead is to store $\mathbf{Auth}_i$, so that fewer nodes have to be computed for $\mathbf{Auth}_{i+1}$. In fact, by observing Auth_4 in Figure 3, one can see that Auth_5 can be trivially computed by letting $a_0 = \mathrm{pk}_4^H$ and keeping the previous values of a_1 and a_2. In the figure, the authentication path is colored in blue, while the nodes that are computed by the verifier are marked in yellow.

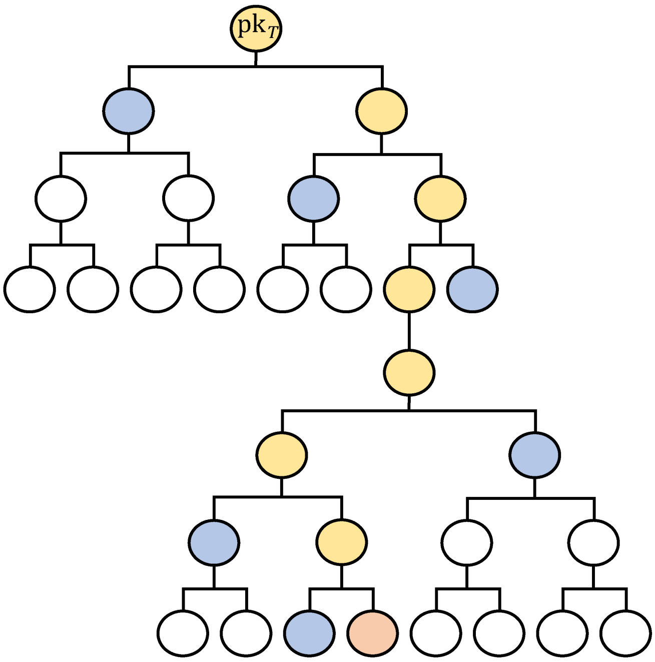
The different techniques to compute $\mathbf{Auth}_i$ are called *Tree Traversal* algorithms, in the literature. We refer to the BDS algorithm (BUCHMANN; DAHMEN; SCHNEIDER, 2008), named after the authors, as the alternative used in XMSS, and a comprehensive survey of different algorithms in (BERNSTEIN; BUCHMANN; DAHMEN, 2008).

3.2.1 Variants

There are at least three main aspects that have been improved over MSS, by modern Merkle Tree HBS schemes. The first is the security assumption of MSS, which relies on collision resistant hash functions. The schemes XMSS and LMS are two internet standards, public available as RFC 8391 (HÜLSING et al., 2018) and RFC 8554 (MCGREW; CURCIO; FLUHRER, 2019), respectively. These schemes rely on SPR, that is a much weaker assumption. This implies that signatures can be much shorter, due to requiring hash functions with half of the output length, to obtain the same security level.

Another aspect is the number of signatures and, consequently, the performance of key generation. Both XMSS and LMS provide a multilevel Merkle Tree structure that greatly improves key generation performance, allowing for much larger number of signatures to be performed under the same public key. In the case of XMSS, this is referred to as the *multi-tree* variant, better known as Multi-Tree eXtended Merkle Signature Scheme (XMSS^{MT}). The idea of the multi-tree is similar to a Public Key Infrastructure (PKI). One can see it as layers of trees where the bottom trees are be used to perform signatures. Then, the trees on layers above sign the root of the tree below them.

Figure 4 – Example of multi-tree with $h = 2$.



Source: The author.

The main benefit of the multi-tree variant is that only the top tree needs to be generated during GEN. The bottom layers are generated on demand, as more OTS key pairs are required. Hence, by allowing multiple levels of trees, this could imply in virtually unlimited number of signatures. The main drawback of such an approach is the increased size of the signatures. This is mainly due to the signatures that link each layer of the tree being part of the finally generated signature. In other words, to verify a signature with a multi-tree, the verifier must verify the signature linking all the layers of the multi-tree, in addition to verifying the OTS signature of the message in the leaf of the bottom tree. See Figure 4, where the authentication path is marked in blue and the nodes computed by the verifier is marked in yellow.

Lastly, we remark that SPHINCS+ (BERNSTEIN et al., 2019) is also an interesting variation and that it is currently under consideration in the NIST standardization process (ALAGIC et al., 2020). This scheme is also a multi-tree scheme, similar to XMSS. However, the leaf nodes use different HBS that allow the same key pair to be used more than once (few times). With this, the main idea is that the leaf node used for the signature is determined pseudo-randomly. The main goal is to eliminate the state condition inherent in HBS, allowing it to be deployed as a general use signature scheme.

In this thesis, we have selected XMSS as a use case, due to its wide acceptability and simplicity in the tree structure. The concept of multi-trees is discussed in our results, as we also aim to reduce key generation. However, we do not include multi-tree benchmarks experiments, since we believe that the increased complexity in the tree structure is unnecessary for the analysis of our results.

4 MESSAGE ENCODING FOR WINTERNITZ ONE-TIME SIGNATURES

In this chapter, we review three variants of the Wots scheme proposed in the literature, with a different approach to compute the encoding of a message M into the t-tuple $\mathcal{B}$ representation. First, we give an overview of the encoding techniques and how the t-tuple may be obtained.

For the last variant, we include a more detailed description of the scheme. This is mainly due to Chapter 6, where we give our main contribution that is heavily based on this alternative. Finally, we give a brief discussion of the different methods here presented and our decision to explore some of these contributions further.

4.1 RUN-LENGTH ENCODING WINTERNITZ

For an arbitrary sequence of binary digits, a run of 0's is the occurrence of one or multiple consecutive 0's enclosed by 1's. Similarly, runs of 1 are enclosed by 0's. For example, let L be a function that returns all the run lengths of a binary string and R be a function that returns the number of runs of the binary string. Then we have that for $M = 00011001$, $L(M) = (b_1, b_2, b_3, b_4) = (3, 2, 2, 1)$ and $R(M) = 4$. However, we observe that the number of runs of M depends on M. Take $M' = 01010101$, then it is easy to see that $R(M) < R(M') = 8$.

This approach has been explored by Steinwandt & VillÁnyi (2008) to build a variant of Wots, which we call Wots-l. The main idea is to use L as the encoding function and R as the checksum. To overcome the issue that $R(M)$ is not fixed, the scheme introduces two new parameters R_{min} and R_{max}, to determine the minimum e maximum number of runs accepted in M. Then we set n to be the largest run-length accepted in $M \in \mathcal{M}^*$ such that

$$\mathcal{M}^* = \{M : M \in \{0,1\}^\ell \wedge R_{min} \leq R(M) \leq R_{max} \wedge \max(L(M)) \leq n\}.$$

With this, we can build an encoding function $\mathbf{E}$ to obtain $\mathcal{B} \in \mathcal{I}_{(t,n)}$ by letting

$$\mathbf{E}(M) = (b_1, \ldots, b_{R(M)}, \ldots, b_{R_{max}}, R(M))$$

where $t = R_{max} + 1$, $(b_1, \ldots, b_{R(M)}) \leftarrow L(M)$ and $b_i = 1$ when $R(M) < i \leq R_{max}$. The first clear difference of this encoding function is that the message space $\mathcal{M}^*$ does not contain all possible binary strings in $\{0,1\}^\lambda$. Hence, we cannot sign any binary message. A second drawback is that the encoding is not a domination free function, that is, any message M and its complement M' produce the same encoding. We can briefly show this with a short example by letting $M = 00011001$ and $M' = 11100110$. Then $\mathbf{E}$ is not domination free since $L(M) = L(M') = (3, 2, 2, 1)$ and $R(M) = R(M') = 4$.

Perhaps the most notable limitation of this proposal comes with the output of L and R. Given that M behaves uniformly at random, we expect $R(M) = \ell/2$ and $n = \log_2 \ell$. Therefore, one cannot select parameters similar to the description of t and w for Wots+ in Section 3.1.2. On the other hand, L is expected to output tuples with small elements, according to Golomb's postulates of randomness (HELLESETH, 2011; GOLOMB; GONG, 2005). That is, we can expect $R(M)/2$ runs to be of length 1, $R(M)/4$ runs of length 2 and so on. This entails that values in $\mathcal{B}$ are also expected to be small. Hence, the run-length encoding can be used to reduce signature cost by decreasing the number of hash iterations required in the signature step of WOTS+, increasing the verification cost as a trade-off. However, (STEINWANDT; VILLÁNYI, 2008) propose to swap the number of iterations that are evaluated following from $\mathcal{B}$ in SIG with VER, so that the verification cost is decreased instead. That is, instead of computing signatures as

$$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_t) = (c_k^{b_1}(x_1, \mathbf{r}), \ldots, c_k^{b_t}(x_t, \mathbf{r})).$$

we can compute them as

$$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_t) = (c_k^{w-1-b_1}(x_1, \mathbf{r}), \ldots, c_k^{w-1-b_t}(x_t, \mathbf{r})).$$

The same applies to the verification step. With this, authors claim 33% faster signature verification if compared to the original `base-w` encoding algorithm with the same signature length.

To conclude, we observe an important technique that inspired one of our contributions in Section 5. To avoid the limitation of $\mathcal{M}^*$, it is proposed to accept any input message $M \in \{0,1\}^\ell$ to the signature algorithm by the inclusion of a randomization step. The idea is to compute $M' = \mathcal{H}_\ell(M||ctr)$, where ctr is a counter variable. The randomization is repeated by iteratively setting $ctr \leftarrow ctr + 1$ and recomputing M' until $M' \in \mathcal{M}^*$ are satisfied, then proceed with the encoding and signature generation.

4.2 NON-ADJACENT FORM ENCODING WINTERNITZ

We discuss in this section, the Non-Adjacent Form (NAF) encoding Winternitz variant (ROH; JUNG; KWON, 2018) called WOTS-N.

Definition 4.2.1. *Let M be an integer. A signed binary representation of M is an equation of the form*

$$m_i = \sum_{i=0}^{\ell-1} m_i 2^i,$$

where $M_i \in \{-1,0,1\}$ for all i. A signed binary representation $(m_{\ell-1}, \ldots, m_0)$ of an integer M is said to be in NAF representation provided no two consecutive m_i's are nonzero.

The main idea proposed in this variant is a `base-w` encoding that takes into account the signed digits of the NAF representation to build $\mathcal{B}$, using the same checksum technique as WOTS+. We do not provide the encoding algorithm, as it is quite extensive. However, the authors claim that the encoding with the checksum results in a domination free function similar to `base-w`. We also remark that a specific algorithm to compute the NAF form of an input message is not given, but as it turns out, there exist different alternatives in the literature (PRODINGER, 2000; OKEYA et al., 2004) that provide different methods to achieve this.

The NAF representation of an integer is unique and similar to that of the binary representation, where $m_i \in \{0, 1\}$. The main advantage of using the NAF representation over the binary representation for Wots based schemes is its low hamming weight. That is, the number of nonzero digits in the NAF representation of an integer is small. For example, the number 3 is represented as (0011) in binary and its NAF representation is $(010-1)$. However for the case of the number 7, its binary representation is (0111) while its NAF representation is $(100-1)$. Additionally, the largest integer represented with 4 digits in NAF form is $10 = (1010)$, and therefore NAF form requires more digits to represent the same range than the binary representation. Hence, the NAF representation can be used to exploit this low hamming weight and produce t-tuples $\mathcal{B}$ with small elements. Then, following the Wots+ definition in Section 3.1.2, signing messages requires less hash iterations. However, as a drawback, the length of the hash chains in the proposed variant are longer than the ones obtained with the original `base-w`. This entails that key generation requires more hash iterations than Wots+. This is also reflected in the signature verification.

In (ROH; JUNG; KWON, 2018), authors achieve 8.5% reduction cost in $C(\textsc{Sig})$ at an exchange of increased 60.9% increase to $C(\textsc{Ver})$. The results are presented with benchmark and security proof under the same assumption as Wots+, which gives a fair comparison under the same security levels and signature size. Lastly, expected costs as well as the benchmarks are only given for $w = 16$, but authors claim that $C(\textsc{Sig})$ can be expected to be lower than Wots+ when $\ell \geq 15 \log_2 w$ and $w \geq 4$

Remark. *Authors in (ROH; JUNG; KWON, 2018) proposed their scheme only considering the case where signature cost is decreased. However, it seems that a similar approach to the run-length encoding is viable. If the number of hash iterations in the signature and verification steps are swapped, then the NAF representation could be used to decrease the number of hash iterations in the verification step instead.*

4.3 CONSTANT-SUM ENCODING WINTERNITZ

In this section, we review a proposal by Cruz, Kaji, and Yatani (CKY) (KAJI; CRUZ; YATANI, 2018), which is a Wots variant with an alternative encoding to `base-w`. Similar to the run-length variant, the number of signature blocks t is no longer determined by the parameter w for this variant. Hence, to avoid confusion, we reintroduce the parameter n that serves a similar purpose as the "chain" length and corresponds to $w-1$ in the Wots+ scheme. Furthermore, most of our contributions are related to this particular scheme. Hence, we provide a more extensive background for this variant.

Let $t, n, s \in \mathbb{N}$ where $n \leq s$ such that

$$\tau_{(t,n,s)} = \left\{(b_1, \ldots, b_t) : 0 \leq b_i \leq n \text{ and } \textstyle\sum_{i=1}^{t} b_i = s\right\},$$

that is, the set of t-tuples whose elements are individually bounded by n and their sum is exactly equal to s. We also define the set $\iota_{(t,n,s)} = \{0, \ldots, |\tau_{(t,n,s)}| - 1\}$. Then, define a constant-sum encoding function $\mathbf{E} : \iota_{(t,n,s)} \to \tau_{(t,n,s)}$, i.e. a bijective map with integer representations of the message space $\mathcal{M} = \{0,1\}^\ell$ in the domain. Notice that it is required that messages are in $\iota_{(t,n,s)}$, i.e., $2^\ell \leq |\tau_{(t,n,s)}|$. Similar to the `base-w` encoding, we prove that the constant-sum encoding is a domination free function in the following lemma.

Lemma 4.3.1. *Let $\mathbf{E} : \iota_{(t,n,s)} \to \tau_{(t,n,s)}$ be a constant-sum encoding, for ℓ, t, n, s positive integers, $n \leq s$ and $2^\ell \leq |\tau_{(t,n,s)}|$. For every $M, M' \in \{0,1\}^\ell$ we have that $\mathbf{E}(M)$ does not dominate $\mathbf{E}(M')$.*

Proof. By definition, we have that $\tau_{(t,n,s)} \in \mathcal{I}_{(t,n)}$. Let $\mathcal{B} = \mathbf{E}(M)$ and $\mathcal{B}' = \mathbf{E}(M')$ and assume that $\mathcal{B}$ dominates $\mathcal{B}'$. Because E is bijective, it follows that $\mathcal{B} \neq \mathcal{B}'$. Therefore, there exists a $1 \leq i \leq t$ such that $b_i > b'_i$. However, since the sum of the vector must be s, then there must be some $1 \leq j \leq t$ where $j \neq i$ and $b_j < b'_j$. Therefore, $\mathcal{B}$ does not dominate $\mathcal{B}'$. □

Let f be some one-way hash function, then we define the variant scheme Wots-cs as follows:

Gen(1^λ). Randomly choose t strings of λ bits to create the secret key $\mathbf{sk} = (x_1, \ldots, x_t)$. The public key is obtained by computing $\mathbf{pk} = \big(f^n(x_1), \ldots, f^n(x_t)\big)$.

Sig$(M, \mathbf{sk})$. Consider an ℓ-bit message $M \in \mathcal{M}$ and its unique constant-sum encoding $\mathbf{E}(M) = (b_1, \ldots, b_t)$. Then, the signature is generated as:

$$\sigma = (\sigma_1, \ldots, \sigma_t) = (f^{n-b_1}(x_1), \ldots, f^{n-b_t}(x_t)).$$

Ver$(M, \sigma, \mathbf{pk})$. Obtain $\mathbf{E}(M) = (b_1, \ldots, b_t)$. The correctness of σ is asserted by computing the remaining iterations over signature blocks and comparing the results with the public key, namely

$$\mathbf{pk} \stackrel{?}{=} (f^{b_1}(\sigma_1), \ldots, f^{b_t}(\sigma_t)).$$

Remark. *The original proposal in (KAJI; CRUZ; YATANI, 2018) and (CRUZ; YATANI; KAJI, 2016) does not use Wots+ construction with c_k function and the $\mathbf{r}$ vector. This is a main drawback to the security of the scheme, as we will discuss further on.*

We point out that a checksum calculation in Sig and Ver is not needed for $\mathbf{E}(M)$. By letting $\mathbf{E}'(M)$ be the `base-w` encoding defined in Section 3.1.3, then $\mathbf{E}(M)$ is analogous to $\mathbf{E}'(M)$ in the sense that tampering with intermediate b_i values while maintaining the constant-sum limitation implies in decreasing a b_i value. Consequently, a malicious party would need to obtain a preimage on iterates of f_k. This occurs naturally as we have proven that $\mathbf{E}$ is a domination free function.

In this variant, the main advantage is that the number of hash iterations performed in Sig and Ver is independent of the signed message. Indeed, the Wots-cs costs $C(\text{Gen}) = tn$, $C(\text{Sig}) = tn - s$ and $C(\text{Ver}) = s$ are precisely determined. As we will see further on, this property can be exploited to choose parameters that always yield reduced cost for the signature-related steps.

4.3.1 CKY constant-sum encoding algorithms and parameters

The original method to compute a constant-sum encoding from an ℓ-bit message (CRUZ; YATANI; KAJI, 2016), which is called CKY-O hereafter, works specifically for the case that "chain" lengths are bounded by s. Thus, $n = s$ and $|\tau_{(t,s,s)}| = \binom{s+t-1}{s}$. The main idea behind the algorithm is to consider the message as an integer $I \in \iota_{(t,s,s)} = \{0, \dots, |\tau_{(t,s,s)}| - 1\}$ and to compute its unique representation by unranking it to $\tau_{(t,s,s)}$. In other words, finding $\mathcal{B} \in \tau_{(t,s,s)}$ such that the rank of $\mathcal{B}$ corresponds to I.

A direct improvement by the same authors (KAJI; CRUZ; YATANI, 2018), which we call CKY-I, explores the fact that tuples in $\tau_{(t,s,s)}$ usually have all b_i distant from their maximum value s. Indeed, if there exists an integer $n << s$ such that $0 \leq b_i \leq n$ and $2^m \leq |\tau_{(t,n,s)}| < |\tau_{(t,s,s)}|$, then costs of signature operations are substantially reduced. Before we can elaborate more on this, we first introduce the required background. The following theorem gives details on the number of such tuples.

Theorem 4.3.2 ((BOLLINGER; BURCHARD, 1990, Item 1d))**.** *Consider the set of t-tuples with elements bounded by n whose sum is s. The cardinality of this set is given by*

$$|\tau_{(t,n,s)}| = \sum_{i=0}^{k} (-1)^i \binom{t}{i} \binom{s-(n+1)i+t-1}{t-1},$$

where $k = \min\left(t, \left\lfloor \frac{s}{n+1} \right\rfloor\right)$.

Proof. Let s be any non-negative integer. The number of ways that t non-negative integers, smaller or equal to n, can be arranged and sum exactly to s, denoted by $|\tau_{(t,n,s)}|$, is given by the coefficient of the term x^s from the polynomial

$$g(x) = (1 + x^1 + \cdots + x^n)^t.$$

We can represent the same polynomial in terms of the sum of the inner geometric series as

$$g(x) = \left(\frac{1-x^{n+1}}{1-x}\right)^t = (1-x^{n+1})^t \frac{1}{(1-x)^t},$$

and from this we expand the binomials such that

$$g(x) = \left(\sum_{i=0}^{t} (-1)^i \binom{t}{i} x^{(n+1)i}\right) \left(\sum_{h=0}^{\infty} \binom{h+t-1}{t-1} x^h\right).$$

However, we are only interested in the coefficient of x^s, which can be expressed as follows:

$$|\tau_{(t,n,s)}| = \sum_{i=0}^{t} (-1)^i \binom{t}{i} \binom{s-(n+1)i+t-1}{t-1}.$$

Finally, the maximum value of i must satisfy $s-(n+1)i+t-1 \geq t-1 \geq 0$ and $t \geq i$. Then, since t is a positive integer, we must have that $s-(n+1)i \geq 0$ only when $i \leq \frac{s}{n+1}$ and $n \geq 0$. Thus,

$k = \min\left(t, \left\lfloor \frac{s}{n+1} \right\rfloor\right)$ and

$$|\tau_{(t,n,s)}| = \sum_{i=0}^{k} (-1)^i \binom{t}{i} \binom{s-(n+1)i+t-1}{t-1}.$$

□

For some input I taken uniformly at random from $\iota_{(t,s,s)}$, the CKY-I algorithm uses the result above observing that $\tau_{(t,n,s)} \subseteq \tau_{(t,s,s)}$, and that $\mathbf{E}(I) \in \tau_{(t,s,s)}$ is also in $\tau_{(t,n,s)}$ with probability $\Pr_{enc} = \frac{|\tau_{(t,n,s)}|}{|\tau_{(t,s,s)}|}$. Thus, after n is fixed, the algorithm proceeds with a trial-and-error procedure to obtain an encoding in $\tau_{(t,n,s)}$. To ensure that it eventually succeeds, for each attempt, the input message is concatenated against a nonce, e.g., $d = \mathcal{H}(M \mid\mid \phi)$, until $\mathbf{E}(d) \in \tau_{(t,n,s)}$. Recall that this is similar to the run-length encoding approach in Section 4.1 and that ϕ can be a counter variable. Algorithm 1 shows the method used in each trial, which degenerates to the deterministic CKY-O when $n = s$.

Algorithm 1 Probabilistic encoding CKY-I

```
Input: $t, n, s \in \mathbb{N}$, $I \in \iota_{(t,s,s)}$
Output: $(b_{t-1}, \ldots, b_0) \in \tau_{(t,n,s)}$
if $t = 1$ then
    return $(s)$
$b \leftarrow 0$
$h_l \leftarrow 0$
$h_r \leftarrow 1$
$a \leftarrow 1$                                  ▷ $a \leftarrow |\tau_{(t-1,0,0)}|$
while not $h_l \leq I < h_r$ do
    $b \leftarrow b + 1$
    $a \leftarrow \frac{a(b+t-2)}{b}$              ▷ $a \leftarrow |\tau_{(t-1,b,b)}|$
    $h_l \leftarrow h_r$
    $h_r \leftarrow h_r + a$
if $s - b > n$ then
    abort
return $(s - b) \mid\mid$ CKY-I$(t-1, n, b, I - h_l)$
```

We now discuss choices of parameters that reduce the cost $C(\text{VER})$ of WOTS-CS when compared to WOTS+. For this task, we search for the appropriate parameters n and s, for a fixed t. The first approach consists of finding the smallest s such that $|\tau_{(t,s-1,s-1)}| < 2^m \leq |\tau_{(t,s,s)}|$, and subsequently choosing the smallest $n \leq s$ such that $2^m \leq |\tau_{(t,n,s)}|$. We call this approach MINVER, since it achieves the smallest $C(\text{VER})$ for WOTS-CS.

This is evidenced in Table 5, where we compare CKY-O and CKY-I using values of t taken from Table 4. For example, when $m = 256$ and $t = 34$, $C(\text{VER})$ is reduced by 34% if compared to the equivalent case for WOTS+ in Table 4. Also, we observe that the case $n < s$ (CKY-I) yields reduced key generation and signature generation costs when compared to the case $n = s$ (CKY-O). This is a great improvement by the authors, however, still more than doubling the cost of signature and key generation in the best case.

Table 5 – Number of iterations of f_k for Wots-cs using MinVer, and success probability of cky-i given as $\Pr_{enc}$. Recall that $C(\text{Ver}) = s$.

ℓ	t	n	$C(\text{Gen})$	$C(\text{Sig})$	$C(\text{Ver})$	$\Pr_{enc}$
256	67	341	22847	22506	341	1.00
		42	2814	2473		~ 0.9792
	45	952	42840	41888	952	1.00
		145	6525	5573		~ 0.9646
	34	2832	96288	93456	2832	1.00
		661	22474	19642		~ 0.9945
512	131	688	90128	89440	688	1.00
		47	6157	5469		~ 0.9756
	89	1849	164561	162712	1849	1.00
		240	21360	19511		~ 0.9994
	66	5855	386430	380575	5855	1.00
		750	49500	43645		~ 0.9907

Source: The author.

4.4 OVERVIEW

We proceed by summarizing the Wots based schemes in Table 6 by their properties, with respect to the choice of parameters, deterministic encoding, domination free encoding function, costs related to the chaining function, and the portability of the encoding function. We observe that the evaluation of portability of the encoding function refers to the practical aspects of implementing it in multiple platforms, including constrained devices with low computation capabilities. We also include our main contribution Wots-cs+ which is presented in Chapter 6, denoted as Wots-cs+. The columns in Table 6 are marked with "✗" to indicate if a property is absent int for each scheme or "✓" otherwise. For cases where the property is not relevant, we have marked with "—".

All variants presented so far propose some improvement to either cost of $C(\text{Sig})$ or $C(\text{Ver})$, by increasing the total length of the hash chains of the scheme, and thus increasing $C(\text{Gen})$. As we have observed previously, the main drawback is that by reducing costs of either $C(\text{Sig})$ or $C(\text{Ver})$, increases the cost of the other in addition to the increased cost to key generation. Moreover, with the exception of Wots-cs and Wots-cs+, whose costs are predetermined and fixed, all variants have costs variations directly related to the bit distribution of the message to be signed.

Another aspect to be evaluated is if the encoding function is domination free. Observe that Wots-l is the only scheme that does not have this property, which gives more credibility to the scheme by avoiding encoding collisions such as the trivial one explained for this same

Table 6 – Properties of different WOTS variants. Costs reduction properties are considered as a comparison to WOTS+.

Property	WOTS+	WOTS-L	WOTS-N	WOTS-CS	WOTS-CS+
Reduces $C(\textsc{Sig})$ or $C(\textsc{Ver})$	—	✓	✓	✓	✓
Fixed $C(\textsc{Sig})$ and $C(\textsc{Ver})$	✗	✗	✗	✓	✓
Domination Free Encoding	✓	✗	✓	✓	✓
Deterministic Encoding	✓	✗	✓	⋆	✓
Flexible Parameters	✓	✗	✓	✓	✓
Portability & Efficiency	✓	✓	✓	✗	✗
Reduces $C(\textsc{Gen})$	—	✗	✗	✗	✓
Security Assumption	SPR	COL	SPR	COL	SPR

Source: The author.

scheme. Moreover, it is the case for WOTS-L and WOTS-CS, that other collisions might appear with the introduction of the randomization technique. Observe that we have marked WOTS-CS with "$\star$" for this property as it allows a deterministic and injective algorithm, at the cost of greatly increasing key and signature generation; see Table 5.

With respect to the parameter selection criteria, WOTS-L seems to be the only variant to have t constrained to the length of the input message ℓ. This is a major drawback, as the run-length encoding function can only be used for large t, and thus limiting its capabilities of reducing signature sizes. For WOTS-N, it is compatible with WOTS+ parameter selections, but it is unclear what are the advantages of this variant for larger w. Notwithstanding, the authors claim that WOTS-N reduces $C(\textsc{Sig})$ for all parameters described for WOTS+ in Table 2, and thus we consider it flexible enough. The WOTS-CS variant is perhaps the most flexible alternative with respect to parameter flexibility. The MINVER criterion allows to choose any pair (n,s) for fixed $t \leq 1$.

For most distinct application scenarios, it is the case that we are interested in performing the entire signing procedure, with the computation of the encoding, in the same device. This is an issue that was not addressed in any of the papers and discuss further in our work. The portability of the encoding function is essential for the general use case of the scheme. However, the constant-sum encoding fails to meet this criterion. Due to the computation of large integer numbers in the encoding process, the algorithm might be impractical when implemented in constrained devices. Nonetheless, considering the case where it can be implemented, the efficiency of the algorithm is questionable. In fact, it is hard to beat `base-w` in this case, as it can be trivially implemented in any platform.

Finally, we address the two last properties that highlight our main contributions with WOTS-CS+. The first novelty we will present in our work is the capability of reducing the overall length of the Winternitz chaining function. That is, for a range of parameters, WOTS-CS+ can decrease $C(\textsc{Gen})$ when compared to WOTS+, and consequently obtain reduced $C(\textsc{Sig})$ and

C(VER) at the same time. For other parameter choices, we show that WOTS-CS+ does not increase C(GEN), and still allows to reduce either C(VER) at the exchange of increasing the cost of the other, or vice versa. Furthermore, we point out that the original papers of the run-length encoding and the constant-sum encoding do not use WOTS+ as a framework for their proposals. Their security proofs rely on stronger assumptions, making it hard to fairly compare the cost trade-offs against other variants, if taking security level into consideration. With the updated description of WOTS-CS, we will show that using our novel deterministic algorithms, we are able to prove security of WOTS-CS+ under SPR assumption. This enables us to give fair comparison to WOTS+, by selecting correct parameters that match signature length and security level.

To conclude, we find that the property of the constant sum is perhaps the most interesting to develop further. The ability to have constant costs for each step seems interesting and the lack of checksum indicates the possibility of reducing the overall cost of the scheme, as we already anticipate here. In the rest of this work, we give our contributions based on two of the variants described in this section. For the first, we take advantage of the randomization idea of WOTS-L to find "better" `base-w` encodings, therefore compensating for the absence of a domination free function in the original work. For the second, we propose different constant-sum encoding algorithms that accept better parameters and even faster encoding speeds, compared to WOTS-CS.

5 TUNING THE WINTERNITZ HASH-BASED DIGITAL SIGNATURE SCHEME

We have seen that both the run-length and constant-sum (CKY-I) use a rehashing technique to achieve faster verification speed, obtaining a message M' that satisfies some requirements. The main drawback is that either the encoding function requires massive computations or the encoding is not a domination free function. In this chapter, we evaluate this technique and its application to the base-w encoding, which is an efficient domination free encoding function. In the following we present our contributions in (PERIN et al., 2018) as published, with some modifications to follow a different notation used in this thesis.

We observe that the original publication was proposed on top of classic WOTS. We present it here as a technique to be used with WOTS+. This is more convenient, since the actual implementations from the paper were extended from the XMSS reference code, which uses WOTS+ and not WOTS. Indeed, our proposals can be extended to either scheme without any modification.

5.1 CHECKSUM PADDING

In this section, we propose to pad unused bits in the Winternitz base-w encoding, reserved for the checksum, with 1. The main goal of this padding is to obtain some performance trade-off, making signature verification faster.

Recall that $Q = \sum_{i=1}^{t_1}(w-1-b_i)$ from Section 3.1.2, line 2. Define Q_{max} and Q_{min} as the greatest and smallest possible values of Q. These situations happen when, $\forall b \in \mathcal{B}_1$, $b = 0$ or $b = w-1$, respectively. Hence,

$$Q_{max} = \sum_{i=1}^{t_1}(w-1-0) = t_1(w-1) \quad \text{and} \quad Q_{min} = \sum_{i=1}^{t_1}(w-1-w-1) = 0.$$

Additionally, the number of bits needed to represent all possible values of Q is given by

$$\mathcal{N}_Q = \lceil \log_2 Q_{max} \rceil = \lceil \log_2 t_1(w-1) \rceil$$

and the number of blocks to accommodate Q is given by t_2, defined in Section 3.1.2.

In general, the number of bits reserved for Q, that is, $t_2 \log_2 w$ bits, is greater than the number of bits actually required for their representation. This difference occurs because an integer number of blocks of size $\log_2 w$ is used to accommodate Q. The number of unused bits is defined as $\mathcal{N}_u = t_2 \log_2 w - \mathcal{N}_Q$.

We note that w and $\mathcal{N}_u$ grow together, as seen in Table 7. This table groups several parameters for WOTS+, how they affect $\mathcal{N}_u$, and presents how these parameter groups benefit from this optimization. Odd powers of 2 for w show no abnormal behavior and are suppressed for simplicity, although there are some combinations where no padding is needed, such as $w = 8, \ell = 128$ and $w = 128, \ell = 512$. Finally, we redefine Q to

$$Q^p = Q + (2^{\mathcal{N}_u} - 1)2^{\mathcal{N}_Q} = Q + 2^{t_2 \log_2 w} - 2^{\mathcal{N}_Q}.$$

In other words, during the signature generation step, we fill the unused $\mathcal{N}_u$ bits with ones. We call the modified scheme WOTS-B.

Table 7 – Unused bits on $\mathcal{B}_2$ for various combinations of w and ℓ.

w	ℓ	$\mathcal{N}_Q$	$\mathcal{N}_u$	$t_2 \log_2 w$	w	ℓ	$\mathcal{N}_Q$	$\mathcal{N}_u$	$t_2 \log_2 w$
4	128	8	0	8	2^{10}	128	14	6	20
	192	9	1	10		192	15	5	20
	256	9	1	10		256	15	5	20
	512	10	0	10		512	16	4	20
16	128	9	3	12	2^{12}	128	16	8	24
	192	10	2	12		192	16	8	24
	256	10	2	12		256	17	7	24
	512	11	1	12		512	18	6	24
64	128	11	1	12	2^{14}	128	18	10	28
	192	11	1	12		192	18	10	28
	256	12	0	12		256	19	9	28
	512	13	5	18		512	20	8	28
256	128	12	4	16	2^{16}	128	19	13	32
	192	13	3	16		192	20	12	32
	256	13	3	16		256	20	12	32
	512	14	2	16		512	21	11	32

Source: The author.

5.1.1 Security Considerations

We remark that classic WOTS uses a padding of zeroes in the signature generation algorithm (MERKLE, 1989). Our proposal based on flipping the padding bits from zeroes to ones moves this fixed amount of iterations of f_k from the verification process to the signature generation. Despite the fact that WOTS+ in RFC 8391 does not take the padding option into consideration, these computations have no impact in security, since they have no checksum purpose, but must be calculated nevertheless.

5.2 TUNING $\mathcal{B}_1$

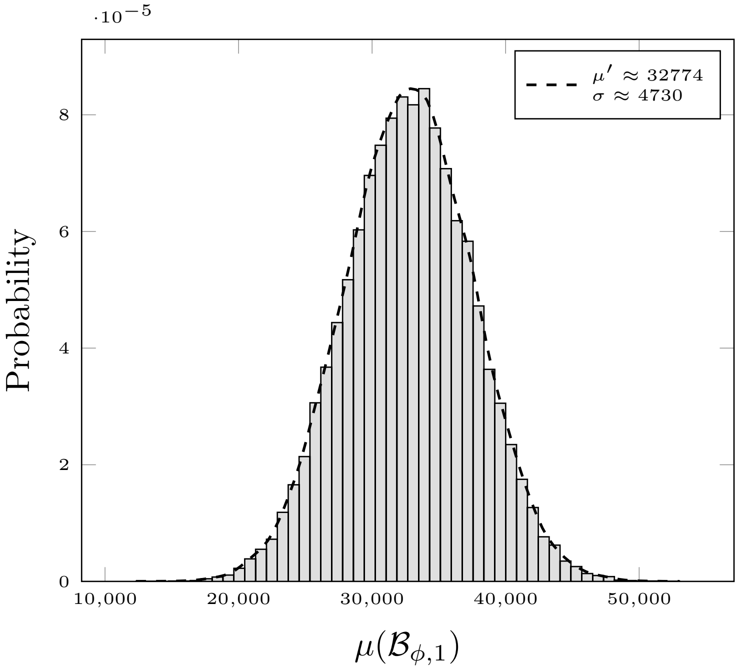
We propose a method to speed up the WOTS+ signature generation or verification without any modification to the original scheme. Our idea is to append a cryptographic nonce to the message to be signed, before generating a signature. We show in Section 5.3 that, by repeating this process and searching for a suitable nonce, we can significantly reduce the cost of the signature verification, in exchange of an increased cost of the signature generation, or vice-versa. In the following, we explain the method and give statistical thresholds for the searching process.

Let M' be a binary message of arbitrary length, R a positive integer and Φ an R-tuple of nonces. We compute values M_ϕ such that $M_\phi = \mathcal{H}_\ell(M' \parallel \phi)$ where $\phi \in \Phi$ and $\mathcal{H}_\ell$ is a cryptographic hash function with output length ℓ. By applying `base-w` encoding to M_ϕ, we are interested in the $\mathcal{B}_1$ vector of the encoding now defined as $\mathcal{B}_{\phi,1} = (b_{\phi,1}, \ldots, b_{\phi,t_1})$. Then let the set of summations of the integer representations of the elements in these tuples be defined by $\mathcal{S} = \{\sum_{b \in \mathcal{B}_{\phi,1}} : \phi \in \Phi\}$. Finally, we choose ϕ from $\min(\mathcal{S})$ to obtain a faster signature generation or from $\max(\mathcal{S})$ for a faster signature verification. Then, we proceed with WOTS+ signature generation SIG described in 3.1.2 by letting $M = M_\phi$ and $\mathcal{B}_1 = \mathcal{B}_{\phi,1}$.

We call the method WOTS-R. Furthermore, this proposal is inspired by (STEINWANDT; VILLÁNYI, 2008), and therefore we observe that Φ could be replaced by the trivial set $\{1, \ldots, R\}$.

5.2.1 Finding a threshold for R

Figure 5 – Normalized histogram of $\mu(\mathcal{B}_{\phi,1})$, with 50 bins and $w = R = 2^{16}$.



Source: The author

In this proposal, we consider R as a statistical parameter that represents the sufficient number of hashes $\mathcal{H}_\ell$ needed to find an adequate summation in $\mathcal{S}$. Intuitively, larger R should produce better results. However, this choice translates to a higher cost for signature generation. We show that there is a suitable threshold for R, depending on the required optimization.

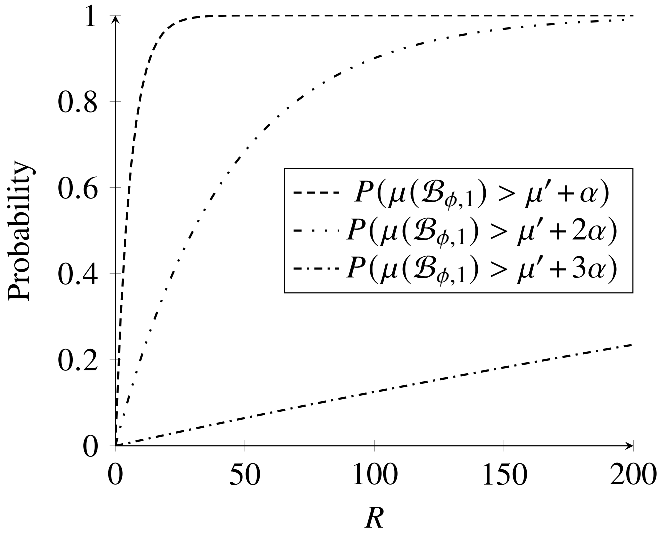
Consider μ as the function that calculates the arithmetic mean of a set of integers. If we assume that $\forall b \in \mathcal{B}_{\phi,1}, 0 \leq b \leq w-1$ follows a uniform distribution, then by repeatedly calculating $\mu(\mathcal{B}_{\phi,1})$, we expect the average $\mu' = \mu(\mu(\mathcal{B}_{\phi,1})) = (w-1)/2$, with $\phi \in \Phi$. Hence, by the central limit theorem, the distribution of the averages $\mu(\mathcal{B}_{\phi,1})$ should follow a normal distribution.

We experiment with $w = 2^{16}$, $m = 256$ and $\mathcal{H}$ = SHA-256. By computing $\mu(\mathcal{B}_{\phi,1})$ with $R = 2^{16}$, indeed we verify that the distribution of the averages follows a normal distribution. Figure 5 is a graphical representation of this behavior. Then, by using the standard normal table (Z-table) with standard deviation α, we have

$$\begin{aligned} P(\mu(\mathcal{B}_{\phi,1}) > \mu' + \alpha) &= 0.1587, \\ P(\mu(\mathcal{B}_{\phi,1}) > \mu' + 2\alpha) &= 0.0228, \\ P(\mu(\mathcal{B}_{\phi,1}) > \mu' + 3\alpha) &= 0.0013. \end{aligned}$$

In Figure 6, we plot the chance of finding $\mu(\mathcal{B}_{\phi,1})$ inside these three intervals, for $R \leq 200$. We use the binomial distribution and distinguish three thresholds as suggestions for values of R: {25, 200, 3500}. Each value yields a probability of 99.9% of finding $\mu(\mathcal{B}_{\phi,1})$ in the intervals previously mentioned, respectively.

Figure 6 – Thresholds for R.



Source: The author.

5.2.2 Security Considerations

Our proposal makes no attempt to modify the underlying classical WOTS+ mechanisms. Hence, we discuss the impact of appending a cryptographic nonce ϕ to the message M' before hashing it. Recall that we must find a suitable hash $M_\phi = \mathcal{H}_\ell(M \mid\mid \phi)$ such that it produces a maximized or minimized sum of $\mathcal{B}_{\phi,1}$. In other words, this introduces a bias, where high-order bits of various blocks in $\mathcal{B}_{\phi,1}$ have a higher probability of being fixed, making partial hash collision attacks more susceptible.

This behavior may be exploited through differential cryptanalysis on the fixed bits for the Merkle-Damgård construction, recently put in practice to generate the first practical collision for SHA-1 (STEVENS et al., 2017). Such an attack could present a threat to our proposal, thus requiring the use of cryptographic hash functions which are second preimage resilient, such as SHA-256. We leave the remaining security considerations for the scheme to be taken from (HÜLSING, 2013b).

5.3 EXPERIMENTS

As a proof-of-concept, we compare the number of iterations of the Winternitz chaining function throughout the entire execution of the digital signature schemes. Again, we denote the four variants tested, Wots+ for the scheme described in Section 3.1.2, Wots-b for the variant described on Section 5.1, Wots-r is the scheme described on Section 5.2 and Wots-br merges the characteristics of the latter two. Considering the discussion on Subsection 5.2.1, sufficient values of R were chosen according to the usual values of the Winternitz parameter w, in addition to $w = 2^{16}$.

In Table 8 we give the average number of iterations of the Winternitz chaining function c_k needed to verify a signature. We experiment with 2^{14} executions of the verification step for the proposed schemes, with $\mathcal{H}$ = SHA-256, $\ell = 256$ and binary messages of 2^{10} bytes generated through `/dev/urandom`. To better understand the effect of each proposal individually, we also compute the average of B_1 and B_2 separately.

Note that $\mu(\mathcal{B}_2)$ is affected "negatively" by Wots-r, since maximizing the sum of elements in $\mathcal{B}_1$ has a direct impact on the calculation of the checksum, minimizing elements in $\mathcal{B}_2$. However, this difference does not heavily impact the overall gains achieved in $\mu(\mathcal{B}_1)$. Furthermore, by using Wots-br, this behavior is mitigated, speeding up the signature generation or verification steps up to a factor of half in a best-case scenario.

In general, we observe a reduction of roughly w iterations of c_k with Wots-b alone. In the case of Wots-r, we obtain a reduction of up to 25% for $w = 16$, 33% for $w = 256$ and 42% for $w = 2^{16}$. By merging both schemes together, we improve these results to 28%, 39 and 52%, respectively.

The aforementioned reductions translate to an increase of similar magnitude on the signature generation. For example, according to Section 3.1.2 and by letting $w = 16$, then $t = 67$ and the total number of iterations of c_k for signature generation and verification is equal to $t(w-1) = 1005$. Our results show that, when $R = 25$ with Wots-br, we can decrease the number of iterations of c_k during the signature verification from approximately 506 to 421 on average.

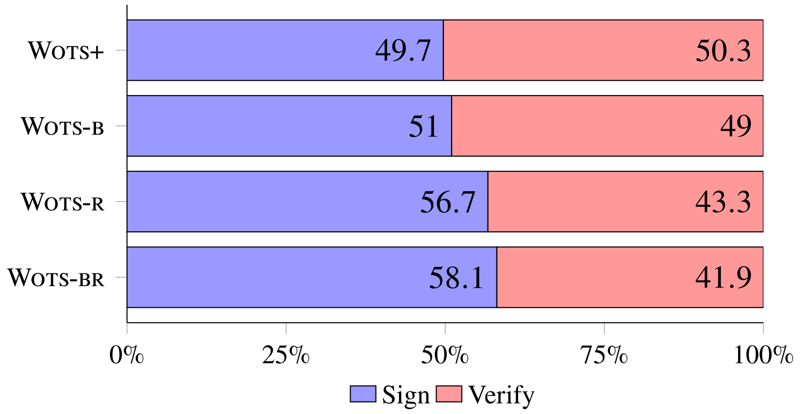
Avoiding these 85 iterations of c_k during the signature verification means that we must now calculate these on the signature generation. In other words, this amounts to a 16.8% speedup for the verification step at the cost of a 15% slowdown during the signature generation, without taking the computation of M_ϕ into account. This trade-off is depicted in Figure 7 and Figure 8

Table 8 – Number of iterations of c_k on the verification step for the proposed schemes when $\max(\mathcal{S})$ is chosen.

w	R	SCHEME	$\mu(\mathcal{B}_1)$	$\mu(\mathcal{B}_2)$	$\mu(\mathcal{B})$
16	-	WOTS+	479.93	25.88	505.80
		WOTS-B		12.25	492.18
	25	WOTS-R	407.81	27.49	435.29
		WOTS-BR		13.49	421.29
	200	WOTS-R	379.08	29.23	408.31
		WOTS-BR		15.23	394.31
	3500	WOTS-R	348.88	31.14	380.02
		WOTS-BR		17.14	366.02
256	-	WOTS+	4081.84	368.43	4450.27
		WOTS-B		136.25	4218.08
	25	WOTS-R	3262.39	370.56	3632.95
		WOTS-BR		130.56	3392.95
	200	WOTS-R	2940.63	372.13	3312.76
		WOTS-BR		132.13	3072.76
	3500	WOTS-R	2604.49	374.75	2979.24
		WOTS-BR		134.75	2739.24
2^{16}	-	WOTS+	525120.63	98231.81	623352.44
		WOTS-B		32707.82	557828.45
	25	WOTS-R	376550.24	98225.54	474775.78
		WOTS-BR		32697.52	409247.76
	200	WOTS-R	319490.02	98850.06	418340.08
		WOTS-BR		33321.91	352811.94
	3500	WOTS-R	262301.92	101022.36	363324.28
		WOTS-BR		35492.57	297794.48

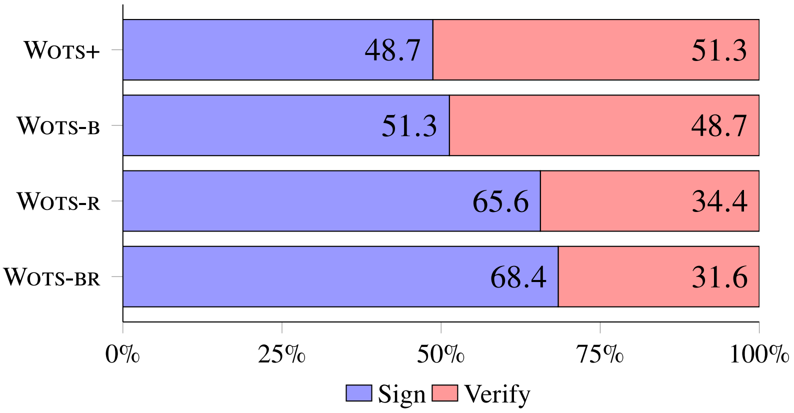
Source: The author.

Figure 7 – Comparing trade-off for current proposals and WOTS+ with $w = 16$ and $R = 25$ and $\max(\mathcal{S})$.



Source: The author.

Figure 8 – Comparing trade-off for current proposals and WOTS+ with $w = 256$ and $R = 3500$ and $\max(\mathcal{S})$.



Source: The author.

We can substantially increase this trade-off by letting $R = 200$ or $R = 3500$, when signature generation time is not constrained. Otherwise, for small values of w, the number of hashes used for WOTS-R might not be an attractive choice. Hence, such values can be better used with greater values of w, where computing hundreds of hash functions is negligible compared to $t(w-1)$.

5.3.1 Impact on Merkle signature schemes

Our proposal has significant results for hash-based schemes that make use of Merkle tree structures. We test WOTS-B, WOTS-R and WOTS-BR with the public domain[1] reference implementation of RFC 8391 XMSS (HÜLSING et al., 2018).

We patch the reference implementation by modifying the padding process inside the WOTS+ signing algorithm, according to Section 5.1, denoting this modification as XMSS-B. Furthermore, by choosing R with the method described in Section 5.2, we achieve up to 32% of speedup when benchmarking the verification step for XMSS. We call this variant XMSS-R. Additionally, when these optimizations are used together, we call the resulting scheme XMSS-BR.

Table 9 shows the average run time of 2^{14} signatures for each scheme, including the computation of R for XMSS-R. We use the recommended value of $w = 16$, and additionally, $w = 256$. For greater values of w (e.g. 2^{16}), it is widely known that the XMSS key generation algorithm is too slow and unpractical. Hence, this value is omitted from the results. Furthermore, we experiment with both $\max(\mathcal{S})$ and $\min(\mathcal{S})$ to demonstrate the impact of our schemes when choosing to optimize signature verification or generation, respectively. In the case of $\min(\mathcal{S})$, XMSS-B and XMSS-BR are not considered, since the default padding of WOTS+ already optimizes signature generation.

Our results in Table 9 show that, for $\max(\mathcal{S})$, any value of R improves the signature verification run time, with the associated cost on signature generation. Evidently, this behavior is

[1] `https://github.com/joostrijneveld/xmss-reference/`

Table 9 – Signature generation and verification run times (in ms) for the proposed schemes.

	w	R	Scheme	Sig. time	Ver. time
$max(S)$	16	-	Xmss	0.953	0.734
			Xmss-b	0.975	0.724
		25	Xmss-r	1.059	0.652
			Xmss-br	1.073	0.637
		200	Xmss-r	1.222	0.620
			Xmss-br	1.240	0.616
		3500	Xmss-r	3.724	0.588
			Xmss-br	3.730	0.573
	256	-	Xmss	7.709	5.676
			Xmss-b	7.908	5.361
		25	Xmss-r	8.597	4.637
			Xmss-br	8.992	4.415
		200	Xmss-r	9.045	4.245
			Xmss-br	9.460	4.052
		3500	Xmss-r	11.760	3.861
			Xmss-br	12.193	3.664
$min(S)$	16	-	Xmss	0.971	0.746
		25	Xmss-r	0.879	0.832
		200		1.006	0.885
		3500		3.393	0.898
	256	-	Xmss	7.819	5.731
		25	Xmss-r	6.672	6.553
		200		6.472	6.982
		3500		8.488	7.435

Source: The author.

Relevant computer specifications are as follows: 8 GB of DDR3 RAM @ 1333MHz, Intel Core i5-4570 @ 3.2GHz and gcc 7.3.0. Base commit for the modifications: 05dac989c40349ad5f4dfee3b563b85131b95332.

suppressed with greater values of w. However, for $\min(\mathcal{S})$, not every value of R may be chosen. In the case of $w = 16$ and $R = 25$, we obtain a speedup of 9.4% for the signature generation process, while when $R = 200$ or $R = 3500$, both processes present a slower run time. The same reasoning applies to $w = 256$, where one should only choose $R = 25$ or $R = 200$.

5.4 DISCUSSION AND FURTHER DEVELOPMENTS

Based on our experimental results, it seems straightforward to assume that the best use case for Wots-br are signature verification. This is mainly because we expect to verify a single signature multiple times, whereas we only compute it once. Also, when signing, run-time is not constrained and thus it is possible to use much greater values for R than 3500 that can further increase verification performance — this is not true if one uses Wots-r to produce faster signatures. More specifically, it might be interesting to use our proposal to produce signatures that are verified more efficiently for IoT devices. Indeed, such are the claims of (BOS et al., 2020), that improved on top of our work to produce XMSS signatures that can be efficiently verified in embedded devices. They give remarkable results using a combination of techniques, including WOTS-R to obtain over a factor of two speed up for signature verification in ARM Cortex-M4 devices. In addition, the same authors propose a better way to integrate WOTS-R into XMSS, and show that the bias introduced in M does not affect the security of XMSS. Furthermore, we observe that the results in (BOS et al., 2020) do not include the WOTS-B variant, as it would break compatibility with RFC 8391.

Another contribution of (BOS et al., 2020) is the analysis of the statistical distribution of $\mathcal{B}_1$ estimated cost of the verification cost for any R that is a power of 2. In the following, we show how $\mathcal{B}_1$ and $\mathcal{B}_2$ behave assuming M is uniform and random, not taking into account the case of WOTS-B or WOTS-R. That is, we describe the expected values of $\mathcal{B}$ for WOTS+, that we find useful to compare expected costs in the coming chapters.

Lemma 5.4.1 (Distribution of $\mathcal{B}_1$ (BOS et al., 2020, Lemma 4))**.** *Fix w and ℓ as positive integers which define $t_1 = \lceil \ell / \log_2 w \rceil$, and let $X = (b_1 + \ldots + b_{t_1})/t_1$ be a random variable; i.e. the mean of the integer values of $\mathcal{B}_1$ where $0 \leq b_i < w$ for $i = 0, \ldots, t_1$. If $M \sim \mathcal{U}_\ell$, then the mean of X, denoted by $\mu(X)$, is equal $(w-1)/2$ and the variance is equal to $(w^2-1)/12t_1$.*

Proof. It follows that if $M \sim \mathcal{U}_\ell$, then $b_i \sim \mathcal{U}_{\log_2 w}$ for $i = 1, \ldots, t_1$, given that $t_1 = \lceil \ell / \log_2 w \rceil$. Therefore $E[b_i] = (w-1)/2$ and $\mathrm{Var}[b_i] = (w^2-1)/12$. Furthermore the values b_i are independent and identically distributed. Then we have that

$$E[X] = E\left[\frac{\sum_{i=1}^{t_1} b_i}{t_1}\right] = \frac{\sum_{i=1}^{t_1} E[b_i]}{t_1} = \frac{t_1 \frac{w-1}{2}}{t_1} = \frac{w-1}{2}$$

and

$$\mathrm{Var}[X] = \mathrm{Var}\left[\frac{\sum_{i=1}^{t_1} b_i}{t_1^2}\right] = \frac{\sum_{i=1}^{t_1} \mathrm{Var}[b_i]}{t_1^2} = \frac{t_1 \frac{w^2-1}{12}}{t_1^2} = \frac{w^2-1}{12 t_1}.$$

□

Indeed, one can verify that our experiment in Figure 5 yield results remarkably close to the expectations in Lemma 5.4.1, thus we can make the following assumption:

Assumption 5.4.2. *The random variable X in Lemma 5.4.1 behaves close to the Normal distribution with mean $\frac{w-1}{2}$ and standard deviation $\sqrt{\frac{w^2-1}{12t_1}}$.*

The hardest claim is to precisely determine the expected value for the checksum values $\mathcal{B}_2$, since they are not independent from $\mathcal{B}_1$. However, since we have that $t_2 << t_1$ for common WOTS+ parameters, we can estimate values for $\mathcal{B}_2$ assuming their expected values are independent as in (BOS et al., 2020; BRUINDERINK; HÜLSING, 2017).

Assumption 5.4.3. *Given $M \sim \mathcal{U}_\ell$, the checksum $\mathcal{B}_2$ of its `base-w` encoding behaves independently and the elements of $\mathcal{B}_2$ follow the uniform distribution.*

This motivates the following lemma:

Lemma 5.4.4 ((BOS et al., 2020, Lemma 5))**.** *Let $Y = b_{t_1+1} + \cdots + b_t$ be a random variable, i.e. the sum of the checksum values of M. Then if $M \sim \mathcal{U}_\ell$, the mean $\mu(Y)$ is equal to $t_2(w-1)/2$ and the variance is equal to $t_2(w^2-1)/12$.*

Proof. The proof follows similar to Lemma 5.4.1. □

We conclude by observing that with these results, we can give approximated average cost of signature generation and signature verification for WOTS+. We have that the average cost of signing $C(\textsc{Sig})$ is approximately equal to the cost of verifying $C(\textsc{Ver})$ for M uniform random. That is,

$$C(\textsc{Sig}) \approx C(\textsc{Ver}) \approx \frac{t(w-1)}{2}$$

and the cost for key generation is

$$C(\textsc{Gen}) = t(w-1).$$

In practice, we have that M is usually the output of some cryptographic function $\mathcal{H}$. If we assume $\mathcal{H}$ behaves like a random function, we obtain the costs for usual parameters of WOTS+ previously described in Table 4.

5.4.1 Errata

It turns out that the equation for t_2 in the original paper could be off for some specific parameters. This caused a miss-calculated result in Table 7 of the paper, when $w = 4$. This has been fixed using of the correct equation of t_2 in Section 3.1.2. The other values for different w remain the same. We thank Antonio Unias for pointing out this issue.

6 IMPROVED CONSTANT-SUM ENCODINGS

In this chapter, we propose improved constant-sum encoding algorithms (PERIN et al., 2021) named WOTS-CS+. Our proposal is motivated by the previous work of CKY in (KAJI; CRUZ; YATANI, 2018; CRUZ; YATANI; KAJI, 2016). Some constant-sum definitions and algorithms have already been presented in Section 4.3. In the following, we define the variant scheme WOTS-CS(λ, ℓ, t, n, s), as follows:

GEN(1^λ). Randomly choose $t+n$ strings of λ bits to create the secret key $\mathbf{sk} = (x_1, \ldots, x_t)$ and $\mathbf{r} = (r_1, \ldots, r_n)$. The public key is obtained by choosing $k \leftarrow \$ \mathcal{K}_\lambda$ and by computing $\mathbf{pk} = \big((\mathbf{r}, k), c_k^n(x_1, \mathbf{r}), \ldots, c_k^n(x_t, \mathbf{r})\big)$.

SIG$(M, \mathbf{sk})$. Consider an ℓ-bit message $M \in \mathcal{M}$ and its unique constant-sum encoding $\mathbf{E}(M) = (b_1, \ldots, b_t)$. Then, the signature is generated as:

$$\begin{aligned}\sigma &= (\sigma_1, \ldots, \sigma_t) \\ &= (c_k^{n-b_1}(x_1, \mathbf{r}) \ldots, c_k^{n-b_t}(x_t, \mathbf{r})).\end{aligned}$$

VER$(M, \sigma, \mathbf{pk})$. Obtain $\mathbf{E}(M) = (b_1, \ldots, b_t)$. The correctness of σ is asserted by computing the remaining iterations over signature blocks and comparing the results with the public key, namely

$$\mathbf{pk} \stackrel{?}{=} ((r, k), c_k^{b_1}(\sigma_1, \mathbf{r}_{n-b_1+1:}), \ldots, c_k^{b_t}(\sigma_t, \mathbf{r}_{n-b_t+1:})).$$

We repeat a previous example given for WOTS+ to illustrate how $\mathbf{r}$ is used to produce the correct public key, now considering the reversed order of the chaining function indexes.

Example. *Let $n = 8$ and consider the signature and verification of a single position i where $b = 3$ from $\mathcal{B}$. Then sign with the $\mathrm{sk}_i = x$ by evaluating $\sigma_i = c_k^{8-3}(x, \mathbf{r}) = c_k^5(x, \mathbf{r})$ and verify the signature by asserting that $\mathrm{pk} \stackrel{?}{=} c_k^3(\sigma_i, \mathbf{r}_{8-3+1:}) = c_k^3(\sigma_i, \mathbf{r}_{6:})$. Observe that the notation of the sub-vector of $\mathbf{r}$ guarantees that c_k starts at the correct position of $\mathbf{r}$, completing the entire hash chain, and thus ensuring the signature verification.*

$$\mathbf{r} = (\underbrace{r_1, r_2, r_3, r_4, r_5}_{\textit{SIG}}, \underbrace{r_6, r_7, r_8}_{\textit{VER}}).$$

In the remaining of the section, we present the results as closely as possible to the original paper. We make some adjustments to preserve the thesis notation.

6.1 PRELIMINARY REMARKS

With the exception of WOTS-R, all encoding alternatives that we have seen so far attempt to reduce $C(\text{VER})$ exclusively. We present an interesting property of the constant-sum encoding that allows flexibility of the encoding parameters and has not been mentioned in previous works.

A result (EGER, 2014) on the symmetry of $|\tau_{(t,n,s)}|$ around $s = \frac{tn}{2}$ implies the interchangeability of $C(\textsc{Sig})$ and $C(\textsc{Ver})$ that does not require any change to the scheme. We restate this result and give a short proof for completion.

Proposition 6.1.1. *Let $n \geq 1$, $t \geq 1$, $0 \leq s \leq tn$. Then,*

$$|\tau_{(t,n,s)}| = |\tau_{(t,n,tn-s)}|.$$

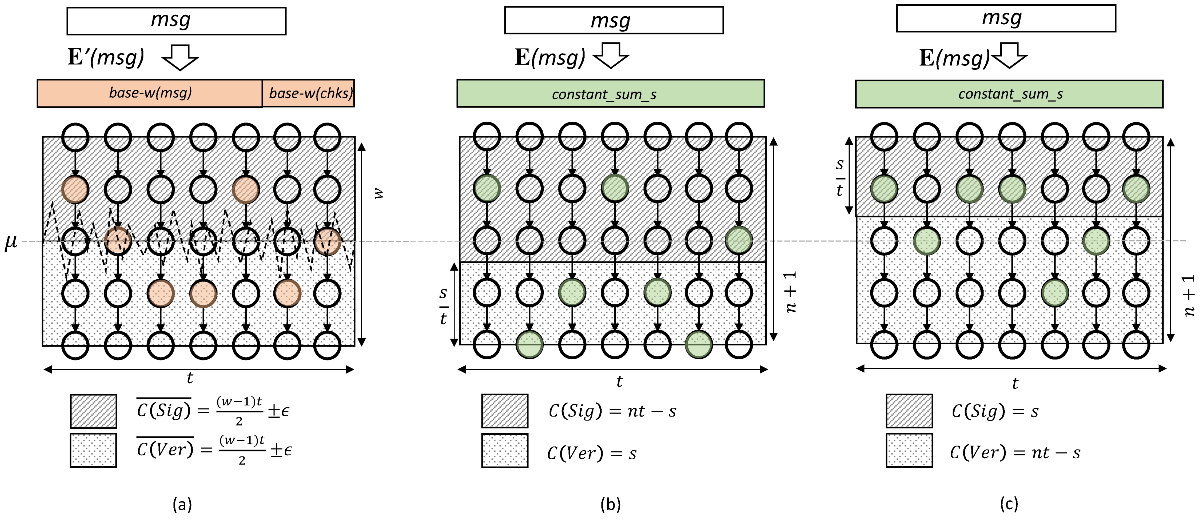
Proof. Consider the bijection

$$\begin{aligned} \mathbf{E}^* &: \tau_{(t,n,s)} \longrightarrow \tau_{(t,n,tn-s)}, \\ \mathbf{E}^*(x_1,\ldots,x_t) &= (n-x_1,\ldots,n-x_t). \end{aligned}$$

We verify that $(x_1,\ldots,x_t) \in \tau_{(t,n,s)}$ if and only if $\sum_{i=1}^{t} x_i = s$ if and only if $\sum_{i=1}^{t}(n-x_i) = tn-s$ if and only if $(n-x_1,\ldots,n-x_t) \in \tau_{(t,n,tn-s)}$. The fact that $\mathbf{E}^*$ is a bijection guarantees that both sets have the same cardinality. □

In other words, we can use the property of the constant sum encoding to exchange the performance of Sig and Ver. In Fig. 6.1, we display how the message encoding affects the hash chains for Sig and Ver in: (a) Wots+, (b) Wots-cs+ with minimization of $C(\textsc{Ver})$, and (c) Wots-cs+ with minimization of $C(\textsc{Sig})$. In each picture, the rectangular areas represent the number of hash chains for Sig and for Ver, which are average values in (a) while exact values in (b) and (c). We observe that minimizing $C(\textsc{Sig})$ has implications on the performance of the encoding function. We will address this in more detail in Section 6.6, as further developments following our published work.

Figure 9 – Illustrative example of the hash operation costs of Wots+ (average) and Wots-cs+ (fixed) for the same t and where $w - 1 = n$ when using MinGen parameter selection strategy.



Source: The author.

Remark. *Proposition 6.1.1 implies that we can use the standard "exponents" of the Wots+ chaining function, instead of the reversed definition of Wots-cs+ in 4.3. For the remainder of* *the thesis, we choose to keep the* Wots-cs+ *notation to preserve the notation in (PERIN et al., 2018).*

6.2 PARAMETER SELECTION STRATEGY MINGEN

We have discussed how CKY proposed parameters using MinVer to obtain Wots-cs signatures that can be verified more efficiently when compared to Wots+. However, as a drawback, these parameters entail a massive increase to $C(\text{Gen})$ and $C(\text{Sig})$. The cky-i algorithm significantly improves on this, but parameters selected with MinVer appear to be non-optimal. We solve these issues by proposing a new approach for parameter selection we call MinGen.

First we give a known binomial identity used in (CRUZ; YATANI; KAJI, 2016) to compute h_r, in Algorithm 1, with $n = s$. The general case where $n \leq s$ is given in the next proposition. This result is mentioned in (FAHSSI, 2012, Table 1), for which we give a short proof.

Proposition 6.2.1. *The cardinality of the set $\tau_{(t,n,s)}$ satisfies*

$$|\tau_{(t,n,s)}| = \sum_{j=0}^{n} |\tau_{(t-1,n,s-j)}|. \tag{6.1}$$

Proof. A t-tuple $\mathbf{b} = (b_1, \ldots, b_t) \in \tau_{(t,n,s)}$ if and only if there exists a $(t-1)$-tuple $(b_2, \ldots, b_t) \in \tau_{(t-1,n,s-b_1)}$. Then we must have that

$$|\tau_{(t,n,s)}| = |\tau_{(t-1,n,s)}| + |\tau_{(t-1,n,s-1)}| + \cdots + |\tau_{(t-1,n,s-n)}|$$

by considering all possible values of b_1, $0 \leq b_1 \leq n$. □

Now we can provide the result below, which states that for fixed t and n, the largest cardinality of $\tau_{(t,n,s)}$ occurs when $s = \lceil \frac{tn}{2} \rceil$ or $s = \lfloor \frac{tn}{2} \rfloor$. We note that $|\tau_{(t,1,s)}| = \binom{s}{t}$, and the case where $|\tau_{(t,n,s)}|$ with $n \geq 1$ is a type of generalization of the binomial coefficients that has been well studied (BOLLINGER; BURCHARD, 1990; EGER, 2014; FAHSSI, 2012). For any fixed $n \geq 1$, together with Proposition 6.1.1, the following theorem implies the unimodality property of $|\tau_{(t,n,s)}|$, for a fixed t. We do not claim the result is new, but in the absence of locating a proof of the exact result, we give an inductive proof.

Theorem 6.2.2. *Let $n \geq 1$, $t \geq 1$ and $0 \leq s \leq \lceil \frac{tn}{2} \rceil$. For any $0 \leq j \leq s$ we have*

$$|\tau_{(t,n,j)}| \leq |\tau_{(t,n,s)}|.$$

Proof. We prove the proposition by induction on t. To simplify calculations, we extend the definition of τ such that $|\tau_{(t,n,s)}| = 0$ for $s < 0$ and $s > tn$. We prove the result for $j \in \mathbb{Z}, j \leq s$. To improve readability, we set $v = \frac{n}{2}$.

The base case is $t = 1$. For any $0 \leq j < s \leq \lceil tv \rceil$, we have $|\tau_{(1,n,s)}| = |\tau_{(1,n,j)}| = 1$, and whenever $j < 0$ we have $|\tau_{(t,n,j)}| = 0$, so the inequality holds for $t = 1$.

For the inductive step, we let $t \geq 2$, $s \leq \lceil tv \rceil$ and assume the main statement holds for $t' = t - 1$, and any j' and s' such that $0 \leq s' \leq \lceil (t-1)v \rceil$ and $j' < s'$. It is enough to show that

$$|\tau_{(t,n,s-1)}| \leq |\tau_{(t,n,s)}|, \tag{6.2}$$

since the repeated application of this equation extends the result for any $j < s$. Recalling that $|\tau_{(t,n,s)}| = \sum_{j=0}^{n} |\tau_{(t-1,n,s-j)}|$ by Eq. (6.1), and letting $\chi = \lceil tv \rceil - s$, we rewrite this equation in terms of χ, which gives

$$|\tau_{(t,n,s)}| = |\tau_{(t,n,\lceil tv \rceil - \chi)}| = \sum_{i=-\lfloor v \rfloor}^{\lceil v \rceil} |\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi + i)}|.$$

The above equality can be verified by carefully analyzing the four cases where n and t assume even or odd values. Now, we write the above equation with $s - 1$ in place of s

$$|\tau_{(t,n,s-1)}| = |\tau_{(t,n,\lceil tv \rceil - \chi - 1)}| = \sum_{i=-\lfloor v \rfloor}^{\lceil v \rceil} |\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi + i - 1)}|.$$

Thus, we get that $|\tau_{(t,n,s-1)}| \leq |\tau_{(t,n,s)}|$ if and only if

$$|\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi - \lfloor v \rfloor - 1)}| \leq |\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi + \lceil v \rceil)}|. \tag{6.3}$$

If $\chi \geq \lceil v \rceil$ then $\lfloor (t-1)v \rfloor - \chi + \lceil v \rceil \leq \lceil (t-1)v \rceil$ and by the induction hypothesis Eq. (6.3) holds. If $\chi < \lceil v \rceil$ we apply Proposition 6.1.1 to find that

$$|\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi + \lceil v \rceil)}| = |\tau_{(t-1,n,\lceil (t-1)v \rceil + \chi - \lceil v \rceil)}|,$$

and thus $\lfloor (t-1)v \rfloor - \chi - \lfloor v \rfloor - 1 \leq \lceil (t-1)v \rceil + \chi - \lceil v \rceil \leq \lceil (t-1)v \rceil$. From this, we apply the induction hypothesis so that

$$\begin{aligned} |\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi - \lfloor v \rfloor - 1)}| &\leq |\tau_{(t-1,n,\lfloor (t-1)v \rfloor - \chi + \lceil v \rceil)}| \\ &= |\tau_{(t-1,n,\lceil (t-1)v \rceil + \chi - \lceil v \rceil)}|. \end{aligned}$$

This completes the proof of Eq. (6.3), that implies Eq. (6.2) and concludes the proof. □

Finally, we describe the MinGen strategy: for fixed t, we let $s = \lceil \frac{tn}{2} \rceil$ and choose the smallest n satisfying $2^\ell \leq |\tau_{(t,n,\lceil \frac{tn}{2} \rceil)}|$. This first step guarantees that we have the smallest n possible for fixed t using Theorem 6.2.2. As a result, the parameter s is no longer constrained to $|\tau_{(t,s-1,s-1)}| < 2^\ell$, and can be substantially larger than the s obtained with MinVer. With t and n fixed, we can now decrease s starting from $\lceil \frac{tn}{2} \rceil$, choosing the smallest value that preserves the condition that $2^\ell \leq |\tau_{(t,n,s)}|$. Even though $C(\text{Ver})$ is not minimal, a clear advantage of this method is that it obtains minimum $C(\text{Gen})$ that consequently reduces $C(\text{Sig})$ and $C(\text{Ver})$ at the same time. In addition, if follows from the symmetry of $\tau_{(t,n,s)}$ described in Proposition 6.1.1 that

Table 10 – Parameters (t, n, s) using MinGen with $30 \leq t \leq 70$.

t	n	s	$C(\text{Gen})$	$C(\text{Sig})$	$C(\text{Ver})$
70	13	375	910	535	375
69	13	427	897	470	427
68	14	402	952	550	402
67	15	400	1005	605	400
66	15	442	990	548	442
65	16	439	1040	601	439
64	17	445	1088	643	445
63	17	531	1071	540	531
62	18	519	1116	597	519
61	19	532	1159	627	532
60	20	556	1200	644	556
59	21	596	1239	643	596
58	23	562	1334	772	562
57	24	603	1368	765	603
56	25	681	1400	719	681
55	27	666	1485	819	666
54	29	687	1566	879	687
53	31	722	1643	921	722
52	33	772	1716	944	772
51	35	862	1785	923	862
50	38	876	1900	1024	876
49	41	935	2009	1074	935
48	45	958	2160	1202	958
47	49	1018	2303	1285	1018
46	53	1117	2438	1321	1117
45	58	1205	2610	1405	1205
44	64	1286	2816	1530	1286
43	70	1474	3010	1536	1474
42	78	1556	3276	1720	1556
41	87	1707	3567	1860	1707
40	98	1835	3920	2085	1835
39	110	2127	4290	2163	2127
38	126	2221	4788	2567	2221
37	144	2490	5328	2838	2490
36	165	2955	5940	2985	2955
35	192	3283	6720	3437	3283
34	226	3643	7684	4041	3643
33	267	4285	8811	4526	4285
32	320	4945	10240	5295	4945
31	388	5790	12028	6238	5790
30	476	6953	14280	7327	6953

Source: The author.

the interchangeability of $C(\textsc{Sig})$ and $C(\textsc{Ver})$ from the previous section preserves the properties of MinGen. In Table 10, we give parameters obtained with MinGen where t is in the range $[30,79]$ minimizing $C(\textsc{Ver})$.

Lastly, we remark that there are limitations in cky-i that prevent the algorithm from taking full advantage of MinGen. As we have discussed, cky-i encodes $\{0,1\}^\ell$ to $\tau_{(t,n,s)}$ by trial-and-error via a mapping to $\tau_{(t,s,s)}$. If $2^\ell \sim |\tau_{(t,n,s)}|$ but 2^ℓ is much smaller than $|\tau_{(t,s,s)}|$, Algorithm 1 is bound to fail with a probability far greater than $\Pr_{enc}$. This happens since Algorithm 1 orders tuples in non-increasing b_{t-1}, so the first 2^ℓ elements of $\tau_{(t,s,s)}$ very likely have $b_{t-1} > n$, and thus, they are not valid tuples in $\tau_{(t,n,s)}$. For example, let $(t,n,s) = (34,226,3643)$ with $\ell = 256$, where $|\tau_{(t,s,s)}| \approx 2^{267} > 2^{256}$. Since $I < 2^{256}$ and $\sum_{j=0}^{s-(n+1)} |\tau_{(t-1,j,j)}| \approx 2^{264} > 2^{256}$ at the end of the `while` loop, we have that $b < s-(n+1)$ which gives $s-b > n+1 > n$, and thus, Algorithm 1 always aborts.

In the next section, we solve these shortcomings by changing the ordering of tuples in Algorithm 1 and providing a deterministic encoding directly into $\tau_{(t,n,s)}$. Moreover, we further explore the MinGen approach in Section 6.4, where we suggest optimal parameters for Wots-cs+.

6.3 DETERMINISTIC ENCODING WITH MINIMAL PARAMETERS

In this section, we propose distinct mapping algorithms that eliminate the probabilistic approach used in cky-i. While our methods are interoperable, they do not produce the same outcome as previous works. Going forward, we show how our algorithms allow for a better choice of parameters regarding costs $C(\textsc{Gen})$, $C(\textsc{Sig})$, and $C(\textsc{Ver})$, by using the MinGen method.

In the following subsections, we present each of our encoding algorithms, but refrain from discussing their relative performance. This is addressed in Section 6.4, in which we give experimental results showing that Algorithms 2 and 3 are considerably faster than both techniques from (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018), for various choices of t, n and s.

6.3.1 Deterministic encoding with minimal parameters

In order to remove the trial-and-error strategy of cky-i, we work on mapping inputs directly to $\tau_{(t,n,s)}$. Our proposal, which is called dcs, is shown in Algorithm 2, and has a total number of iterations bounded by $t(n+1)$.

The correctness of dcs is assured directly from Proposition 6.2.1. Let $\tau^{(h)}_{(t,n,s)} = \{(j,b_2,\dots,b_t) \in \tau_{(t,n,s)} : 0 \le j \le h\}$ for $0 \le h \le n$. Then, $|\tau^{(h)}_{(t,n,s)}| = \sum_{j=0}^{h} |\tau_{(t-1,n,s-j)}|$, i.e. the partial sum up to h from Eq. (6.1). We give a simplified explanation next. Any value $I \in \iota_{(t,n,s)}$ is bounded by $|\tau^{(h')}_{(t,n,s)}| \le I < |\tau^{(h'+1)}_{(t,n,s)}|$ with $-1 \le h' < n$ and $|\tau^{(-1)}_{(t,n,s)}| = 0$. Hence, $b_1 = h'+1$ and

Algorithm 2 Deterministic constant-sum encoding DCS

```
Input: $t, n, s \in \mathbb{N}$, $I \in \iota_{(t,n,s)}$
Output: $(b_1, \ldots, b_t) \in \tau_{(t,n,s)}$
if $t = 1$ then
    return $(s)$
$b \leftarrow 0$
$h_l \leftarrow 0$
$h_r \leftarrow |\tau_{(t-1,n,s)}|$
while not $h_l \leq I < h_r$ do
    $b \leftarrow b + 1$
    $h_l \leftarrow h_r$
    $h_r \leftarrow h_r + |\tau_{(t-1,n,s-b)}|$
return $(b) \,||\, \text{DCS}(t-1, n, s-b, I-h_l)$
```

the algorithm continues recursively with new parameters $(I - |\tau^{(h')}_{(t,n,s)}|) \in \iota(t-1, n, s-h'+1)$, until all elements of the tuple are determined.

Example. *Let* $I = 15$ *and* $(t,n,s) = (4,2,3)$. *Then it follows that* $|\tau^{(1)}_{(4,2,3)}| = 13$ *and* $|\tau^{(2)}_{(t,n,s)}| = 16$. *Hence, we must have that* $b_1 = 2$*, since* $13 \leq I < 16$.

$$(\underbrace{\overset{b_1=0}{0,1,2,3,4,5,6,}}_{|\tau_{(3,2,3)}|=7}\ \underbrace{\overset{b_1=1}{7,8,9,10,11,12,}}_{|\tau_{(3,2,2)}|=6}\ \underbrace{\overset{b_1=2}{13,14,\mathbf{15}}}_{|\tau_{(3,2,1)}|=3}).$$

The recursive step takes $I = 2$ *and* $(t,n,s) = (3,2,1)$.

$$(\ \underbrace{\overset{b_2=0}{0,1,}}_{|\tau_{(2,2,1)}|=2}\ \ \underbrace{\overset{b_2=1}{\mathbf{2}}}_{|\tau_{(2,2,0)}|=1}\ )$$

Since we have $\mathcal{B} = (2,1,b_3,b_4)$ *and the sum already equals s, the following steps return* b_3 *and* b_4 *equal to zero, thus* $\mathbf{E}(I) = (2,1,0,0)$.

Our method allows for the same parameter space as exemplified in Table 5. However, there is a significant improvement when comparing CKY-I and DCS with regard to a different parameter strategy. Since Algorithm 2 does not use $|\tau_{(t,s,s)}|$ to compute h_l, we can search for parameters that yield minimal $C(\text{Gen})$ by drastically decreasing n, using the MinGen method. These parameters and associated costs are discussed in Section 6.4 and detailed in Table 11.

Nevertheless, this advantage comes at a price, as we are now challenged with the task of computing large binomials given from Proposition 6.2.1, which uses Theorem 4.3.2 repeatedly for every iteration of the `while` loop. This elevated complexity was avoided in CKY-I by reusing values previously assigned to h_r; see line 11 of Algorithm 1.

6.3.2 Binary search encoding

Due to the differences between Algorithm 1 and Algorithm 2 described previously, the latter is relatively more complex to compute as a result of repeated large binomial calculations. It is desirable to perform the underlying unranking algorithm more efficiently since constant-sum encodings are calculated in every signature generation and verification.

The task of searching for the aforementioned boundary h by individually computing and summing terms of Eq. (6.1) constrains previous algorithms to a linear number of iterations. To efficiently compute $|\tau^{(h)}_{(t,n,s)}|$, we manipulate the binomial coefficients used to evaluate $|\tau_{(t,n,s)}|$ through Proposition 6.2.1 by expanding the technique from Theorem 4.3.2.

Proposition 6.3.1. *For any* $0 \le h \le n$,

$$|\tau^{(h)}_{(t,n,s)}| = \sum_{i=0}^{k} (-1)^i \binom{t-1}{i} \times \left[\binom{s-(n+1)i+t-1}{t-1} - \binom{s-(n+1)i+t-2-h}{t-1} \right],$$

where $k = \min\left(t, \left\lfloor \frac{s}{n+1} \right\rfloor\right)$.

Remark. *In the proof below, we use the* Column-sum *property of the pascal's triangle that yields the following identity*

$$\sum_{j=0}^{\alpha} \binom{j}{\beta} = \binom{\alpha+1}{\beta+1}.$$

Proof. We recall that the summation in Theorem 4.3.2 has an upper bound k due to the first binomial coefficient evaluating to zero for values $i > k$. Then, from Proposition 6.2.1 and by letting

$$k = \max_{j\in\{0,\ldots,n\}} \left(\min\left(t, \left\lfloor \frac{s-j}{n+1} \right\rfloor \right) \right) = \min\left(t, \left\lfloor \frac{s}{n+1} \right\rfloor\right),$$

we can express the cardinality $|\tau^{(h)}_{(t,n,s)}|$ as

$$\sum_{j=0}^{h} \sum_{i=0}^{k} (-1)^i \binom{t-1}{i} \binom{s-(n+1)i+t-2-j}{t-2}.$$

By exchanging the order of the summations, we have

$$|\tau^{(h)}_{(t,n,s)}| = \sum_{i=0}^{k} (-1)^i \binom{t-1}{i} \left[\sum_{j=0}^{h} \binom{s-(n+1)i+t-2-j}{t-2} \right].$$

For any $0 \le j \le h$, we let $\alpha = s-(n+1)i$ and $\beta = t-2$. Then, the inner summation can be simplified as:

$$\begin{aligned} \sum_{j=0}^{h} \binom{\alpha+\beta-j}{\beta} &= \sum_{j=0}^{\alpha+\beta} \binom{j}{\beta} - \sum_{j=0}^{\alpha+\beta-h-1} \binom{j}{\beta} \\ &= \binom{\alpha+\beta+1}{\beta+1} - \binom{\alpha+\beta-h}{\beta+1}. \end{aligned}$$

Substituting the values of α and β yields the proof. □

Proposition 6.3.1 allows us to compute $|\tau^{(h)}_{(t,n,s)}|$ directly, for any h. From this, we maintain the rationale of the previous algorithm, but remove its recursive approach and additionally employ a binary search strategy to determine the boundary h. Our new method, called DBCS, is given in Algorithm 3, with the total number of iterations at most $t\log_2(n+1)$.

Algorithm 3 Binary search constant-sum encoding DBCS

```
Input: $t, n, s \in \mathbb{N}$, $I \in \iota_{(t,n,s)}$
Output: $\mathcal{B} = (b_1, \ldots, b_t) \in \tau_{(t,n,s)}$
$\mathcal{B} \leftarrow (0, \ldots, 0)$
for $1 \leq i \leq t$ do
    $c \leftarrow \min(n, s)$
    $b \leftarrow 0$
    while $c > 0$ do
        $step \leftarrow \left\lfloor \frac{c}{2} \right\rfloor$
        $j \leftarrow b + step$
        if $|\tau^{(j)}_{(t-i,n,s)}| \geq I$ then
            $j \leftarrow j + 1$
            $b \leftarrow j$
            $c \leftarrow c - step + 1$
        else
            $c \leftarrow step$
    $b_i \leftarrow b$
    if $b > 0$ then
        $I \leftarrow I - |\tau^{(b-1)}_{(t-i,n,s)}|$
        $s \leftarrow s - b$
return $\mathcal{B}$
```

6.3.3 Encoding as part of the signature

As seen in the previous sections, the tuples resulting from our deterministic algorithms are not trivially computable. This impacts the performance of operations with signatures. With the intent of reducing such costs, we propose a modification to WOTS-CS+ which replaces the encoding calculation in VER, increasing its performance even further, at the expense of increased signature sizes.

We suggest two adjustments to the underlying scheme. In SIG, the encoding $\mathcal{B}$ is sent as part of σ. This requires the signature size to be incremented by $t\lceil \log_2(n+1) \rceil$ bits. In VER, we propose to use Algorithm 4, hereafter denoted as VCS, to assert the correspondence of I and $\mathcal{B}$ without searching for boundaries, which could be costly due to the large binomial calculations.

Algorithm 4 Constant-sum encoding verification VCS

```
Input: $t, n, s \in \mathbb{N}$, $I \in \iota_{(t,n,s)}$, $\mathcal{B} \in \tau_{(t,n,s)}$
Output: $\mathbf{E}(I) \overset{?}{=} \mathcal{B}$
for $1 \leq i \leq t$ do
    if not $|\tau^{(b_i-1)}_{(t-i,n,s)}| \leq I < |\tau^{(b_i)}_{(t-i,n,s)}|$ then
        return false
    $I \leftarrow I - |\tau^{(b_i-1)}_{(t-i,n,s)}|$
    $s \leftarrow s - b_i$
return true
```

Remark. *As Proposition 6.3.1 requires* $0 \leq h \leq n$*, we observe that in the case that* $b_i = 0$*, we admit* $|\tau_{(t-i,n,s)}^{(b_i-1)}| = 0$.

The method VCS is essentially a simplified version of Algorithm 3, since it only checks that the boundary conditions calculated in SIG are correct. If any of the t elements in the tuple $\mathcal{B}$ does not lie in the correct interval given by the implementation of $\mathbf{E}$, the signature verification fails. We note that the modifications in the signature structure proposed to use VCS do not make it easier for a malicious party to create a forgery since the encoding is a bijective map, and thus a unique representation of the original message. The detailed security claims of the scheme, however, are presented in Section 6.5.2.

6.3.4 Memoization of intermediate sums

A common technique of dynamic programming may be employed in the methods above, namely DCS, DBCS and VCS, in the form of a lookup table containing intermediate values $|\tau_{(\cdot,\cdot,\cdot)}|$. This creates a memory trade-off that hinders the portability of the algorithm to constrained devices, but other situations may benefit from the sharp performance increase. We denote these variant encodings as DCS-M, DBCS-M and VCS-M, respectively.

Storage requirements are in the order of $s(t-1)$ integers with up to ℓ bits each for DCS-M. This comes directly from Eq. (6.1), in which every value from the range $|\tau_{(\gamma,n,\delta)}|$ with $1 \leq \gamma \leq t-1$ and $1 \leq \delta \leq s$ may be needed, and thus stored. The resulting lookup table is partially shaped like an upper triangular matrix. For $0 < i < \lfloor \frac{s}{n} \rfloor$, the i-th row has exactly $n \times i$ integers, and other rows are completely filled with s values.

In the case of DBCS-M and VCS-M, a three-dimensional matrix of order $s(t-1)(n+1)$ is needed, composed of ℓ-bit integers. In this case, we have access to all partial sums from the range $|\tau_{(\gamma,n,\delta)}^{(\zeta)}|$ with $0 \leq \zeta \leq n$. The resulting matrix is also partially upper triangular. We note that these requirements may be optimized to reduce storage demands and be used as typical implementation strategies. However, depending on the choice of parameters and encoding algorithms, the storage requirement may become impractical.

For instance, we estimate the worst case scenario without optimization for parameters given in Table 11. We assume that each stored value is an arbitrary precision integer with at most ℓ-bits. Then, if we consider $\ell = 256$ and $t = 67$, we require roughly 0.8 MB for DCS-M and 13.5 MB for DBCS-M and VCS-M. Now, if we let $t = 34$ for the same ℓ, the memory requirement increases to 3.8 MB and 870 MB, respectively. We discuss the trade-off associated with these estimates in Section 6.4, along with our benchmark results.

6.4 PARAMETER CHOICE AND BENCHMARKING

This section shows how the choice of parameters and techniques for the constant-sum encoding impacts the efficiency of WOTS-CS+. The performance of such algorithms is measured via the number of applications of f_k needed for the entire scheme, as well as their running time. Additionally, previous work in the literature (CRUZ; YATANI; KAJI, 2016) shows results computed with cryptographic hash functions that are currently considered insecure, such as SHA-1 (LEURENT; PEYRIN, 2020). We thus consider updated security parameters in our results.

For $\ell = 256$ and $\ell = 512$, we search combinations of parameters bounded by $30 \leq t \leq 80$ and $60 \leq t \leq 140$, respectively, such that $\ell \lesssim \log_2 |\tau_{(t,n,s)}|$. From this space, according to the MinGen approach, we select values of n and s that minimize costs of Gen for each value of t. Table 11 shows the results of the search grouped into usual values of t, where the cost prepended with a Δ represents changes in relation to Wots+.

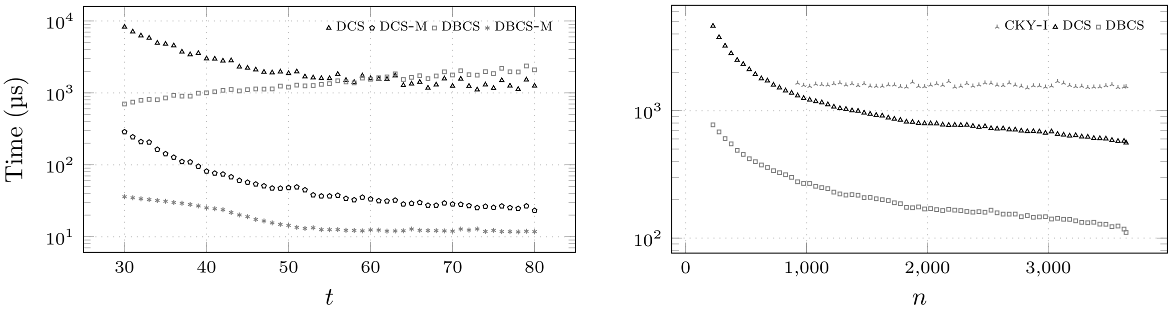
Table 11 – Suggested parameters for Wots-cs+ using MinGen for a given t as compared to Wots+.

ℓ	t	n	C(Gen)	C(Sig)	C(Ver)	ΔC(Gen)
256	67	15	1005	605	400	+0.00%
	45	58	2610	1405	1205	−7.94%
	34	226	7684	4041	3643	−11.37%
512	131	15	1965	1120	845	+0.00%
	89	57	5073	2695	2378	−9.52%
	66	241	15906	8227	7679	−5.49%

Source: The author.

The dominant term t in the signature size has direct implications on the number of iterations of f_k. A trade-off happens if t is increased, to the extent that the values n and s may be reduced. In the literature, parameters are given to obtain an efficient signature verification. For sufficiently small values of t, e.g. $t = 34$ for $\ell = 256$ and $t = 66$ for $\ell = 512$, we highlight that, by applying the MinGen approach, we minimize C(Gen) and obtain *at the same time* increased performance of Gen, Sig and Ver, when compared to Wots+.

Figure 10 – Comparison of running time between distinct constant-sum encoding techniques, in microseconds, for $\ell = 256$.



(a) Our proposals with $n \neq s$ and parameters minimizing C(Gen).

(b) Encoding to $\tau_{(34,n,3643)}$ with $n \in \{226, 276, \ldots, 3626, 3643\}$.

Source: The author.

It is also fundamental to discuss the cost of encoding algorithms. In order to perform a fair comparison with the original methods, we give running time results of each algorithm performing encodings in the context of the aforementioned parameter space. We implement CKY-I, DCS, DBCS, VCS and memoized variants using the GMP C++ library. The resulting software is available at `github.com/zambonin/wots-cs` and compiled using `clang` 11.0.0 with `-O3` optimization level. Times are measured considering an AMD Ryzen™ 3300X, running at 3.8GHz.

Figure 10 shows the running time of the different methods considered. The y-axis is presented in a logarithmic scale for better readability. We give a twofold discussion as follows, remarking that the `base-w` computation is absent from the comparisons due to its negligible running time. Figure 11a depicts a comparison of the different techniques proposed in Section 6.3 and Section 4.3, with sets of parameters as calculated in Table 11. The relative performance of Algorithms DCS and DBCS changes with the size of the target encoding tuples.

The lower complexity of the binary search is more evident when the search space n is large. Such behavior is expected due to the higher cost of computing $|\tau^{(h)}_{(t,n,s)}|$ defined in Proposition 6.3.1. Algorithm DBCS outperforms DCS when $t < 58$, that is, it is much faster for smaller values of t. For example, encoding into tuples of the set $\tau_{(34,226,3643)}$ is $\approx 82.7\%$ faster when compared with DCS

We also give remarks on the running time of VCS. Recalling that $\mathcal{B}$ must be available during VER, then this algorithm can be used to verify the correspondence of the signed message with $\mathcal{B}$. We refrain from showing it in Fig. 10, since it is significantly faster to use this approach, instead of repeatedly encoding the signed message with any of the other proposed algorithms. Indeed, it essentially removes the cost associated with the encoding.

Figure 11b shows the average time to encode a digest into a tuple with fixed t and s for different values of n. For this experiment, we also consider the original probabilistic encoding CKY-I, which fails to yield encodings for lower values of n, the most interesting cases due to savings in $C(\text{GEN})$. This is expected, due to its limitation when $|\tau_{(t,s,s)}|$ is much larger than 2^{ℓ}, as seen in Section 6.2. Disregarding these sets of parameters, we find that DCS and DBCS are, on average, $\approx 46\%$ and $\approx 87\%$ faster than CKY-I, respectively. We also experimentally find that for sets of parameters in Table 11, at least DBCS is again consistently faster.

6.4.1 Results in Merkle signature schemes

We now demonstrate how WOTS-CS+ performs in the context of a Merkle-based signature scheme, observing the cost related to all variations of the encoding algorithms that we have proposed so far. We compare instantiations of XMSS with WOTS+ in the leaf nodes, in contrast to the usage of WOTS-CS+. We modify the existing XMSS reference implementation from RFC 8391 (HÜLSING et al., 2018) to use our algorithms in consonance with this goal.

Table 12 and Table 13 shows running times of distinct combinations of the encoding algorithms for all steps of the signature scheme, called $C_x(\text{GEN})$, $C_x(\text{SIG})$ and $C_x(\text{VER})$ for

Table 12 – Performance of different encoding algorithms in the context of `XMSS_SHA2_10_256`, in tens of millions of CPU cycles ($\times 10^7$), for $t = 67$. Values in square brackets are the percentage of the cost dedicated solely to encoding.

Encoding		$t = 67$	
algorithm(s)	C_x(Gen)	C_x(Sig)	C_x(Ver)
`base-w`	155	0.53 [0.006%]	0.08 [0.029%]
DCS	158	0.79 [29.86%]	0.31 [76.78%]
DBCS	155	0.86 [36.08%]	0.38 [81.32%]
DCS + VCS	157	0.79 [30.28%]	0.20 [65.77%]
DBCS + VCS	161	0.88 [35.15%]	0.20 [64.89%]
DCS-M	153	0.60 [7.122%]	0.11 [37.52%]
DBCS-M	148	0.60 [10.67%]	0.13 [48.07%]
DCS-M + VCS-M	151	0.59 [7.250%]	0.07 [5.028%]
DBCS-M + VCS-M	152	0.62 [10.78%]	0.07 [2.967%]

Source: The author.

Table 13 – Performance of different encoding algorithms in the context of `XMSS_SHA2_10_256`, in tens of millions of CPU cycles ($\times 10^7$), for t = 34. Values in square brackets are the percentage of the cost dedicated solely to encoding.

Encoding		$t = 34$	
algorithm(s)	C_x(Gen)	C_x(Sig)	C_x(Ver)
`base-w`	1163	4.12 [0.001%]	0.58 [0.005%]
DCS	1061	4.62 [18.10%]	1.35 [62.34%]
DBCS	1027	3.83 [3.448%]	0.61 [21.08%]
DCS + VCS	1039	4.55 [18.41%]	0.52 [6.288%]
DBCS + VCS	1075	3.99 [3.318%]	0.53 [5.916%]
DCS-M	1051	3.81 [1.055%]	0.53 [6.789%]
DBCS-M	1032	3.77 [0.754%]	0.51 [5.040%]
DCS-M + VCS-M	1059	3.82 [1.065%]	0.49 [0.716%]
DBCS-M + VCS-M	1038	3.75 [0.712%]	0.49 [0.653%]

Source: The author.

brevity, and additionally highlights the encoding cost for signature operations. The underlying XMSS tree has been configured with a height parameter of $h = 10$ and SHA-256 as the chaining function, i.e. the RFC parameter `XMSS_SHA2_10_256`. Measurements are done in the same CPU specification as in the previous section and were repeated 2^3 times for C_x(Gen) and 2^{3+h} for C_x(Sig) and C_x(Ver), with the average running time shown in the respective columns. We focus on the parameters $\ell = 256$ with $t = 67$ and $t = 34$, as given by Tables 4 and 11, with the goal of minimizing C_x(Gen).

In the case of $t = 67$, the behavior shown in Fig. 11a is evidenced with DCS being slightly faster than DBCS. Minimal differences between all algorithms are observed with regard to C_x(Gen), due to the fact that $w - 1 = n$ from Table 11 and Table 4. The most interesting claim, in this case, is the effect of encoding execution times. Gains to C_x(Ver) are only observed for two cases: DCS-M + VCS-M and DBCS-M + VCS-M. Fortunately, for these parameters, the memory requirement for the memoization is significantly lower when compared to the next case, thus viable for implementations that are not memory constrained.

We recall that the use of VCS entails an increased signature size. When $t = 67$, $|\sigma|$ is increased by 34 bytes, i.e., 67 four-bit integers. To make a fair comparison with WOTS+, we would need to use $t = 66$, which implies a similar signature size and $\approx 1.5\%$ improvement on $C_x(\text{GEN})$, as per Table 10. However, since $C_x(\text{VER})$ is also increased, we do not find this an interesting avenue to pursue. With regard to $t = 34$, we achieve significantly reduced $C_x(\text{GEN})$. In fact, with the exception of DCS, all other combinations of algorithms provide faster overall running time execution. For this case, it is important to evaluate two distinct aspects.

The memory cost for the algorithms DBCS-M or VCS-M may reach roughly 870 MB in the worst case, limiting their usefulness only for very specific devices where memory is abundant. When using DCS-M, the memory cost reaches nearly 4 MB, which is tolerable in exchange for the extra performance gain in a wider range of platforms. Moreover, we cannot provide a reasonable parameter set if $t = 33$, to make up for the 34 extra bytes in the signature when using VCS. Therefore, we highlight that algorithm DBCS still achieves better overall costs, with no extra memory cost and no increment to the signature size.

Overall, we show that there exist sets of parameters where the constant-sum encoding complexity is counterbalanced so that the scheme is efficient. It should also be noted that $t = 34$ is not the most commonly used parameter for WOTS+ instances. However, it is useful to demonstrate our results, and it has applications in the case of smaller signature requirements that we will discuss in Section 7.1.

Finally, the experiment shows that some practical limitations restrain us from reaching theoretical improvement estimates. For example, we observe that both DCS and DBCS use a large portion of the verification running time. In fact, this fraction may even increase if we consider implementations that are more optimized for verification running time than the XMSS reference code. For this reason, we believe that more extensive research of constant-sum encoding could provide optimizations that may compensate for this encoding cost, making the scheme more competitive. Alternatively, we discuss, in the following subsection, some applications that can avoid the impact of the encoding costs.

6.4.2 Alternative parameter choices

To show the benefit of reduced chain lengths for signature generation, in spite of higher encoding costs, it is interesting to point out that not every application will require the message encoding to be performed by the same device that produces the signature. It is common, for signature applications, to perform all cost-intensive computations outside the secure platform that holds the private key. One clear example of this is the analogous case where large documents are digested into small hash outputs before being sent to a secure device in order to be signed, e.g., a smart card, a trusted platform module (TPM) or a hardware security module (HSM).

Some of these applications might benefit from a signature scheme that can be configured to have a small key generation cost and a small signature generation cost. For such cases, the cost of the signature verification may be less critical. Hence, we propose the use of the same parameters in Table 11, but exchanging the costs of signature and verification using Proposition 6.1.1. In Table 14, we display the alternative parameters with the changes in relation to WOTS+ regarding the cost of signature generation.

Table 14 – Suggested parameters for WOTS-CS+ using MINGEN for a given t as compared to WOTS+.

ℓ	t	n	$C(\text{GEN})$	$C(\text{SIG})$	$C(\text{VER})$	$\Delta C(\text{SIG})$
256	67	15	1005	400	605	−20.39%
	45	58	2610	1205	1405	−14.99%
	34	226	7684	3643	4041	−15.96%
512	131	15	1965	845	1120	−13.99%
	89	57	5073	2378	2695	−15.17%
	66	241	15906	7679	8227	−8.74%

Source: The author.

If we carefully examine WOTS-CS+, there is a clear advantage on the number of hash operations performed during the signing process for all parameter choices. We highlight the 20.39% reduced cost for a signature generation when $t = 67$. For this particular case, there is a balanced trade-off in the verification time, with 20.39% more hash functions in the verification, when compared to WOTS+. For the other cases, all costs related to GEN, SIG and VER are reduced (compare Table 4 with Table 14).

We note that this performance gain may be less effective when used with XMSS. For the same parameters used in the previous benchmark, we obtain XMSS signature costs reduced by only 2–3% when $t = 67$ (excluding the encoding cost). This is mainly attributed to the predominant cost of key generation, necessary to build the Merkle tree verification path during the signature. In fact, depending on the application, it is possible to optimize the verification path construction by caching tree nodes, such that the key generation cost is mitigated, thus achieving greater performance gains closer to the ones stated in Table 14.

For $t = 34$, the improvements are more noticeable. We can use the same parameters in Table 13 and obtain a 15% improvement that is closer to the 15.96% described in Table 14, since there is an additional 11.37% improvement for every key generated during the construction of the verification path. In addition, this parameter selection also presents 11% faster key generation and roughly 13% faster key verification (excluding the encoding cost).

Last, we find that it would be more interesting to study the practical results of our improvements with experiments that are tailored for each application. There are several different ways to combine our parameters with different optimization strategies available for XMSS, considering distinct computation and memory capabilities. We leave such studies for consideration in future works.

6.5 SECURITY PROOF OF WOTS-CS+

In (KAJI; CRUZ; YATANI, 2018), the authors claim to have proved that WOTS-CS is existentially unforgeable under an adaptive chosen-message attack (EU-CMA), assuming that the constant-sum encoding function is collision resistant and f_k is one-way collision resistant. However, we find that their estimation of certain probabilities is incorrect. More specifically, the authors make incorrect implicit assumptions about the uniformity of the probability distribution of tuples controlled by the adversary.

In order to avoid such assumptions, we provide a security proof that WOTS-CS+ is EU-CMA by closely following the argument in (HÜLSING, 2013b) and a recent revision of the same security proof in (KUDINOV; KIKTENKO; FEDOROV, 2020). Several results on the behavior of $|\tau_{(t,n,s)}|$ are needed to obtain the correct probability bounds of our proof.

In this section, we provide brief security definitions, necessary properties of the distribution of $\tau_{(t,n,s)}$ and a detailed proof of WOTS-CS+ under the assumption that $\mathcal{F}_\lambda$ is undetectable and second-preimage resistant one-way function. This result is a major improvement on the previous works by CKY, since the proof does not require collision resistance. In particular, we can instantiate WOTS-CS+ with signatures about half the size when compared to (KAJI; CRUZ; YATANI, 2018). Moreover, we give a concrete bound of the security level of our scheme, showing that its security level is comparable with WOTS+.

6.5.1 Properties of $\tau_{(t,n,s)}$

We briefly introduce two properties of $\tau_{(t,n,s)}$ in the form of Proposition 6.5.1 and Proposition 6.5.2. These results are imperative for the characterization of the probability assertions we make in Theorem 6.5.3, providing the security bounds of WOTS-CS+.

We recall the definition of the constant-sum encoding and address its distribution probability with respect to the choice of parameters and the size of the signed message. Remark that $\mathbf{E} : \iota_{(t,n,s)} \longrightarrow \tau_{(t,n,s)}$, as introduced in Section 4.3, with $\iota_{(t,n,s)} = \{0, \ldots, |\tau_{(t,n,s)}| - 1\}$ and $2^\ell \leq |\tau_{(t,n,s)}|$. Let us call $\tau^\ell_{(t,n,s)}$ the image of $\{0,1\}^\ell$ under $\mathbf{E}$, namely the evaluation of $\mathbf{E}(M)$, allowing a slight abuse of notation as we naturally identify $\{0, \ldots, 2^\ell - 1\}$ with $\{0,1\}^\ell$. Because the image of $\mathbf{E}(M)$ is $\tau^\ell_{(t,n,s)}$ instead of $\tau_{(t,n,s)}$, we must ensure that 2^ℓ is very close to $|\tau_{(t,n,s)}|$, so that we can use results about distribution of tuples in $\tau_{(t,n,s)}$, when considering such a distribution in $\tau^\ell_{(t,n,s)}$.

Proposition 6.5.1. *Let ℓ and t be integers such that $t = o(\ell)$, and let n be such that*

$$|\tau_{(t,n-1,\lfloor\frac{(n-1)t}{2}\rfloor)}| < 2^\ell \leq |\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|.$$

Let $\tau^\ell_{(t,n,\lfloor\frac{nt}{2}\rfloor)}$ be the image of the encoding $\mathbf{E}(\{0,1\}^\ell) \subseteq \tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}$. Consider the sequence of independent and uniformly distributed random variables $X = X_\ell$ from $\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}$, and the *sequence of independent and uniformly distributed random variables $Y = Y_\ell$ from $\tau^{\ell}_{(t,n,\lfloor\frac{nt}{2}\rfloor)}$. Then, X and Y are statistically indistinguishable.*

Proof. We need to prove that for every $D > 0$ there exists ℓ_0, such that:
$\frac{1}{2}\sum_{\alpha\in\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}} |\Pr[X_\ell = \alpha] - \Pr[Y_\ell = \alpha]| < \frac{1}{\ell^D}$, for all $\ell \geq \ell_0$.

The left-hand side of the inequality is equal to

$$\begin{aligned} & \frac{1}{2}\left(\sum_{\alpha\in\tau^{\ell}_{(t,n,\lfloor\frac{nt}{2}\rfloor)}} \left|\frac{1}{2^\ell} - \frac{1}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|}\right| + \sum_{\alpha\in(\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}\setminus\tau^{\ell}_{(t,n,\lfloor\frac{nt}{2}\rfloor)})} \left|\frac{1}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|} - 0\right|\right) \\ = {} & 1 - \frac{2^\ell}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|} = \frac{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| - 2^\ell}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|}. \end{aligned} \tag{6.4}$$

From (EGER, 2014, Eq. (5)), we know that $|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| \sim \frac{1}{\sqrt{\frac{\pi t}{2}}}\frac{(n+1)^t}{\sqrt{(n+1)^2-1}}$. So, for ℓ, n large enough, our hypothesis translates to

$$\frac{1}{\sqrt{\frac{\pi t}{6}}}\frac{n^t}{\sqrt{n^2-1}} < 2^\ell \leq \frac{1}{\sqrt{\frac{\pi t}{6}}}\frac{(n+1)^t}{\sqrt{(n+1)^2-1}},$$

Therefore,

$$\begin{aligned} 2^\ell & \leq |\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| \leq \frac{1}{\sqrt{\frac{\pi t}{6}}}\frac{(n+1)^t}{\sqrt{(n+1)^2-1}} \\ & = \frac{1}{\sqrt{\frac{\pi t}{6}}}\frac{n^t}{\sqrt{n^2-1}}\frac{(n+1)^t}{n^t}\frac{\sqrt{n^2-1}}{\sqrt{(n+1)^2-1}} \leq 2^\ell\left(1+\frac{1}{n}\right)^t\frac{\sqrt{n^2-1}}{\sqrt{n^2+2n}}. \end{aligned}$$

Using $|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| \leq 2^\ell\left(1+\frac{1}{n}\right)^t\frac{\sqrt{n^2-1}}{\sqrt{n^2+2n}}$ to bound the numerator and using $2^\ell \leq |\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|$ to bound the denominator of the left-hand side of (6.4), we obtain

$$\frac{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| - 2^\ell}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|} \leq \frac{2^\ell\left(\left(1+\frac{1}{n}\right)^t\frac{\sqrt{n^2-1}}{\sqrt{n^2+2n}} - 1\right)}{2^\ell} < \left(1+\frac{1}{n}\right)^t - 1.$$

Expanding the binomial in the right-hand side, we obtain

$$\frac{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}| - 2^\ell}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|} < 1 + \frac{t}{n} + \frac{\binom{t}{2}}{n^2} + \cdots + \frac{1}{n^t} - 1.$$

Since $2^\ell \leq \frac{1}{\sqrt{\frac{\pi t}{6}}}\frac{(n+1)^t}{\sqrt{(n+1)^2-1}}$, we have $2^{\ell/t} \leq \frac{1}{(\frac{\pi t}{6})^{1/(2t)}}\frac{(n+1)}{(n^2+2n)^{1/(2t)}}$. Thus, we get $\frac{\binom{t}{i}}{n^i} \leq \frac{\binom{t}{i}}{2^{\ell/t}}\frac{n+1}{n^i}\frac{1}{(\frac{\pi t}{6})^{1/(2t)}(n^2+2n)^{1/(2t)}}$. Since $t = o(\ell)$, then $\frac{\binom{t}{i}}{2^{\ell/t}}$ approaches zero faster than any polynomial in ℓ. In particular, for a given positive constant D, there exists $\ell_0^{(i)}$ such that $\frac{\binom{t}{i}}{n^i} \leq \frac{\binom{t}{i}}{2^{\ell/t}}\frac{n+1}{n^i}\frac{1}{(\frac{\pi t}{6})^{1/(2t)}(n^2+2n)^{1/(2t)}} < \frac{1}{t\ell^D}$, for all $\ell \geq \ell_0^{(i)}$. Now, considering $\ell_0 = \max\{\ell_0^{(1)}, \ell_0^{(2)}, \ldots, \ell_0^{(t)}\}$, we get $\frac{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|-2^\ell}{|\tau_{(t,n,\lfloor\frac{nt}{2}\rfloor)}|} < 1 + \frac{t}{n} + \frac{\binom{t}{2}}{n^2} + \ldots + \frac{1}{n^t} - 1 \leq \frac{1}{\ell^D}$, for all $\ell \geq \ell_0$. □

Proposition 6.5.1 guarantees that taking (λ,ℓ,t,n,s) such that $\ell = \mathrm{poly}(\lambda)$, $t = o(\ell)$, $s = \frac{tn}{2}$ and n such that

$$|\tau_{(t,n-1,\frac{(n-1)t}{2})}| < 2^{\ell} \leq |\tau_{(t,n,\frac{nt}{2})}|$$

yields acceptable parameters, as defined in Definition 6.5.1 in Section 6.5.2, since $X = X_\ell$ and $Y = Y_\ell$ are statistically indistinguishable. This ensures that parameters chosen in our experiments using MinGen are acceptable and satisfy the requirements for the security of the scheme.

The following proposition gives a probability bound used in the security proof given in Theorem 6.5.3.

Proposition 6.5.2. *Let $t,n \geq 2$, $0 \leq \eta \leq n$ and $s \leq \frac{tn}{2}$. Let $(b_{t-1},\ldots,b_0)$ be chosen uniformly at random from $\tau_{(t,n,s)}$. Then, for any $0 \leq j \leq \eta$ and for any $0 \leq i \leq t-1$,*

$$\frac{|\tau_{(t-1,n,s-\eta)}|}{|\tau_{(t,n,s)}|} \leq \Pr[b_i = j] \leq \frac{|\tau_{(t-1,n,s)}|}{|\tau_{(t,n,s)}|}.$$

Proof. We observe that b_i follows the same distribution as b_0 for any i. This holds since for any t-tuple in $\tau_{(t,n,s)}$, each possible index permutation of the tuple is in $\tau_{(t,n,s)}$. Furthermore, we know that $|\tau_{(t-1,n,s-j)}|$ is the cardinality of the set of all t-tuples in $\tau_{(t,n,s)}$ where $b_0 = j$. Hence,

$$\Pr[b_i = j] = \Pr[b_0 = j] = \frac{|\tau_{(t-1,n,s-j)}|}{|\tau_{(t,n,s)}|}.$$

Applying Theorem 6.2.2, we get the bounds

$$\frac{|\tau_{(t-1,n,s-\eta)}|}{|\tau_{(t,n,s)}|} \leq \Pr[b_i = j] \leq \frac{|\tau_{(t-1,n,s)}|}{|\tau_{(t,n,s)}|}.$$

□

6.5.2 Security proof

We now prove the security of Wots-cs+ through the following theorem, which closely follows the arguments in (HÜLSING, 2013b) and a more recent revision of the same proof framework (KUDINOV; KIKTENKO; FEDOROV, 2020). We remark that the proof is given only for constant-sum encoding algorithms compatible with parameters given in Definition 6.5.1.

Definition 6.5.1. *For (growing) security parameter λ the parameters (λ,ℓ,t,n,s) are* acceptable *if $\ell = \mathrm{poly}(\lambda)$, $t \geq 2$, $n \geq 2$, $2n-1 \leq s \leq \frac{tn}{2}$, $2^\ell \leq |\tau_{(t,n,s)}|$ and the uniform distribution $X = X_\ell$ on $\tau_{(t,n,s)}$ is computationally indistinguishable from the uniform distribution $Y = Y_\ell$ on $\tau^{\ell}_{(t,n,s)}$.*

Theorem 6.5.3. *Let (λ,ℓ,t,n,s) be acceptable Wots-cs+ parameters where λ is the security parameter. Let $\mathcal{F}_\lambda = \{f_k : \{0,1\}^\lambda \to \{0,1\}^\lambda, k \in \mathcal{K}_\lambda\}$ be a one-way, second preimage resistant and undetectable function family. The insecurity of Wots-cs+ scheme against an EU-CMA* *attack satisfies*

$$\begin{aligned}\mathrm{InSec}^{\text{EU-CMA}}(\textit{Wots-cs+}(\lambda,\ell,t,n,s);z,1)\\ \leq \frac{t\cdot|\tau_{(t,n,s)}|}{|\tau_{(t-1,n,s-(n-1))}|}\Big(n\cdot\mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_\lambda;\tilde{z})\\ +\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda;z')+n\cdot\mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda;z')\Big),\end{aligned}$$

with $\tilde{z}=z+3tn+n$ and $z'=z+3tn$ and where time is given in number of evaluations of the function f_k in $\mathcal{F}_\lambda$.

Proof. Assume, for the sake of contradiction, that there exists an adversary $\mathcal{A}$ that can produce existential forgeries for Wots-cs+(λ,ℓ,t,n,s) by running an adaptive chosen message attack in time at most z and with success probability $\epsilon_{\mathcal{A}}\equiv\mathrm{Succ}^{\text{EU-CMA}}_{\text{Wots-cs+}(\lambda,\ell,t,n,s)}(\mathcal{A})$ that is greater than the insecurity bound claimed by the theorem. Then, we construct an oracle machine $C^{\mathcal{A}}$ that either breaks OW or SPR properties of $\mathcal{F}_\lambda$ using $\mathcal{A}$.

We let $\tilde{\epsilon}_{\mathcal{A}}$ be the probability that $C^{\mathcal{A}}$ (given in Algorithm 5) arrives at line 22 by calling $\mathcal{A}$. That is

$$\tilde{\epsilon}_{\mathcal{A}}=\Pr[b_\alpha\leq\beta\wedge\text{“Forgery is valid”}\wedge b'_\alpha>\beta],$$

where “Forgery is valid” is 1 if and only if $\textsc{Ver}(M',\sigma',\mathbf{pk})$ succeeds and $M'\neq M$, or 0 otherwise.

From line 22 and forward, there are two possibilities:

1. either $\beta=0$ or the intermediate steps of the verification of σ'_α contains a preimage of y_c. Then, $C^{\mathcal{A}}$ returns preimage x with probability 1.

2. $\beta>0$ and the intermediate steps of the verification of σ'_α do not contain a preimage of y_c. Then, $C^{\mathcal{A}}$ returns a second preimage for x_c, if the chaining function over σ'_α collides with the chaining function of y_c at position $n-\gamma$. Since we have picked γ uniformly at random in $\{0,\ldots,\beta-1\}$, this event occurs with probability $\frac{1}{\beta}$ lower bounded by $\frac{1}{n}$.

Then, by considering that the first case occurs with probability p and the other with probability $1-p$, we use the OW and SPR assumptions to upper bound $\tilde{\epsilon}_{\mathcal{A}}$ as

$$p\cdot\tilde{\epsilon}_{\mathcal{A}}\leq\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda;z')$$

for the first case, and

$$(1-p)\frac{\tilde{\epsilon}_{\mathcal{A}}}{n}\leq\mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda;z')$$

for the second. The new $z'=z+3tn$ is an upper bound of the time $C^{\mathcal{A}}$ takes to setup, sign and call $\mathcal{A}$. Finally, we sum both equations and obtain

$$\tilde{\epsilon}_{\mathcal{A}}\leq\mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda;z')+n\cdot\mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda;z'). \tag{6.5}$$

In the remainder of the proof, we derive a lower bound on $\tilde{\epsilon}_{\mathcal{A}}$ as a function of $\epsilon_{\mathcal{A}}$, considering the possibility that $\mathcal{A}$ might be able to distinguish the challenges inserted by $C^{\mathcal{A}}$.

Algorithm 5 $C^{\mathcal{A}}$

Input: λ, $k \in \mathcal{K}$ and challenges $y_c, x_c \in \{0,1\}^{\lambda}$.

Output: A value x that is either a preimage of $f_k(y_c)$ or a second preimage of $f_k(x_c)$ or fail.

```
 1: $(\mathbf{sk}, \mathbf{pk}) \leftarrow \textsc{Gen}(1^{\lambda})$.
 2: $\alpha \leftarrow\$ \{1, \dots, t\}$
 3: $M_0 \leftarrow\$ \{L \in \{0,1\}^{\ell} : (a_1, \dots, a_t) = \mathbf{E}(L) \wedge a_\alpha \neq n\}$
 4: $a = (a_1, \dots, a_t) \leftarrow \mathbf{E}(M_0)$
 5: $\beta = a_\alpha$                                        ▷ Note that $\beta \neq n$
 6: **if** $\beta = 0$ **then**
 7:     $\mathbf{r}' \leftarrow \mathbf{r}$
 8: **else**
 9:     $\gamma \leftarrow\$ \{0, \dots, \beta - 1\}$
10:     $\mathbf{r}' = \mathbf{r}$ and replace $r'_\gamma = c_k^{\beta-\gamma-1}(y_c, \mathbf{r}_{n-\beta+1:}) \oplus x_c$.
11: Obtain $\mathbf{pk}'$ where $\mathbf{pk}'_{\mathbf{i}} = c_k^n(\mathbf{sk}_{\mathbf{i}}, \mathbf{r}')$ for $1 \leq i \leq t, i \neq \alpha$ and $\mathbf{pk}'_\alpha = c_k^{\beta}(y_c, \mathbf{r}'_{n-\beta+1:})$.
12: Receive $M$ from $\mathcal{A}$ in the query phase and $(b_1 \dots, b_t) \leftarrow \mathbf{E}(M)$
13: **if** $b_\alpha > \beta$ **then**
14:     **return** fail
15: $\sigma = \textsc{Sig}(M, \mathbf{sk}, \mathbf{r}')$
16: Replace $\sigma_\alpha = c_k^{\beta - b_\alpha}(y_c, \mathbf{r}'_{n-\beta+1:})$
17: Reply to query with $\sigma$ and obtain $(M', \sigma')$ from $\mathcal{A}$
18: **if** $(M', \sigma')$ is a valid signature **then**
19:     Let $(b'_1, \dots, b'_t) \leftarrow \mathbf{E}(M')$
20:     **if** $b'_\alpha \leq \beta$ **then**
21:         **return** fail
22:     **else if** $\beta = 0$ **or** $c_k^{b'_\alpha - \beta}(\sigma'_\alpha, \mathbf{r}'_{n-b'_\alpha+1:}) = y_c$ **then**
23:         **return** preimage $x = c_k^{b'_\alpha - \beta - 1}(\sigma'_\alpha, \mathbf{r}'_{n-b'_\alpha+1:}) \oplus \mathbf{r}_{n-\beta}$
24:     **else if** $x = c_k^{b'_\alpha - \gamma - 1}(\sigma'_\alpha, \mathbf{r}'_{n-b'_\alpha+1:}) \oplus \mathbf{r}_{n-\gamma} \neq x_c$
        **and** $c_k^{b'_\alpha - \gamma}(\sigma'_\alpha, \mathbf{r}'_{n-b'_\alpha+1:}) = c_k^{\beta-\gamma}(y_c, \mathbf{r}'_{n-\beta+1:})$ **then**
25:         **return** second preimage $x$
26: **return** fail
```

In this case, $\mathcal{A}$ may intentionally provide a message M such that $b_\alpha > \beta$ or a forgery where $b'_\alpha \leq \beta$, thus making the probability $\tilde{\epsilon}_{\mathcal{A}}$ much smaller than $\epsilon_{\mathcal{A}}$.

Let two distributions $\mathcal{D}_C$ and $\mathcal{D}_{\textsc{Gen}}$ over $\{0, \dots, n-1\} \times \{0,1\}^{\lambda} \times \{0,1\}^{\lambda \times n} \times \mathcal{K}_\lambda$. A sample $\{\beta, u, r, k\} \in \mathcal{D}_C$ is equivalent to choosing β as described in lines 2–5 of $C^{\mathcal{A}}$, and u, r, k uniformly at random. A sample $\{\beta, u, r, k\} \in \mathcal{D}_{\textsc{Gen}}$ corresponds to choosing β as described in lines 2–5 of $C^{\mathcal{A}}$, choosing r, k uniformly at random and $u = c_k^{n-\beta}(x, r)$ with $x \leftarrow\$ \{0,1\}^{\ell}$. That is, $\mathcal{D}_C$ corresponds to the elements generated by $C^{\mathcal{A}}$, while $\mathcal{D}_{\textsc{Gen}}$ corresponds to the elements generated in one Wots-cs+ signature chain up to $n - \beta$.

Let $C'^{\mathcal{A}}$ be a new algorithm that is similar to $C^{\mathcal{A}}$, but takes input from $\mathcal{D}_C$ or $\mathcal{D}_{\textsc{Gen}}$ where $y_c = u$, see Algorithm 6. Then, $C'^{\mathcal{A}}$ behaves as $C^{\mathcal{A}}$ up until line 20, where if condition $b'_\alpha \leq \beta$ is not satisfied, returns 1, and otherwise returns 0. Hence, given inputs from $\mathcal{D}_C$, $C'^{\mathcal{A}}$ outputs 1 with probability $\tilde{\epsilon}_{\mathcal{A}}$, identical to $C^{\mathcal{A}}$. However, given inputs from $\mathcal{D}_{\textsc{Gen}}$, the

Algorithm 6 $C'^{\mathcal{A}}$

```
Input: $\lambda$, sample $(\beta, u, \mathbf{r}, k)$
Output: 0 or 1
(sk, pk) ← Gen($1^\lambda$).                    ▷ $r$ and $k$ are taken from sample instead
$\alpha \leftarrow\$ \{1, \dots, t\}$
Obtain $\mathbf{pk}'$ where $\mathbf{pk}'_{\mathbf{i}} = c_k^n(\mathbf{sk}_\mathbf{i}, \mathbf{r}')$ for $1 \le i \le t, i \ne \alpha$ and $\mathbf{pk}'_\alpha = c_k^\beta(y_c, \mathbf{r}'_{n-\beta+1:})$.
Receive $M$ from $\mathcal{A}$ in the query phase and $(b_1 \dots, b_t) \leftarrow \mathbf{E}(M)$
if $b_\alpha > \beta$ then
    return 0
$\sigma$ = Sign($M, \mathbf{sk}, \mathbf{r}'$)
Replace $\sigma_\alpha = c_k^{\beta - b_\alpha}(y_c, \mathbf{r}'_{n-\beta+1:})$
Reply to query with $\sigma$ and obtain $(M', \sigma')$ from $\mathcal{A}$
if $(M', \sigma')$ is a valid signature then
    Let $(b'_1, \dots, b'_t) \leftarrow \mathbf{E}(M')$
    if $b'_\alpha \le \beta$ then
        return 0
    else
        return 1
return 0
```

probability that $C'^{\mathcal{A}}$ outputs 1 is

$$\hat{\epsilon}_{\mathcal{A}} \equiv Pr[b_\alpha \le \beta \wedge \text{“Forgery is valid”} \wedge b'_\alpha > \beta]$$

with $Pr[\text{“Forgery is valid”}] = \epsilon_{\mathcal{A}}$. This holds, since, in this case, $\mathcal{A}$ receives information from a legitimate Wots-cs+ signature, whereas in the previous case $\mathcal{A}$ receives tampered data. Then, we can rewrite the previous equation as

$$\begin{aligned} &\epsilon_{\mathcal{A}} \cdot \Pr[b_\alpha \le \beta \wedge b'_\alpha > \beta \mid \text{“Forgery is valid”}] \\ &\qquad\qquad \ge \epsilon_{\mathcal{A}} \cdot \Pr[b_\alpha = \beta \wedge b'_\alpha > b_\alpha \mid \text{“Forgery is valid”}]. \end{aligned}$$

Now we may evaluate each probability of the right-hand side of the inequality individually. Let us consider a random variable $X = |\{i : 1 \le i \le t, b_i < n\}|$ under the condition “Forgery is valid”. That is, the number of elements in $\mathbf{E}(M)$ at the query phase of $C'^{\mathcal{A}}$ that are not equal to n and produce a valid forgery if σ is sent to $\mathcal{A}$. Then, recalling that $\alpha \leftarrow\$ \{1, \dots, t\}$, by Proposition 6.5.2 taking $j = \beta$ and $\eta = n - 1$, we have

$$Pr[b_\alpha = \beta \mid \text{“Forgery is valid”}] \ge \frac{X}{t} \cdot \frac{|\tau_{(t-1, n, s-(n-1))}|}{|\tau_{(t,n,s)}|}.$$

Furthermore, due to the properties of the constant-sum encoding, we must have that at least one element in $\mathbf{E}(M')$ has increased when compared to $\mathbf{E}(M)$. These elements then must be at least one among all X elements, and thus by picking α uniformly at random we have that

$$Pr[b'_\alpha > b_\alpha \mid b_\alpha = \beta \wedge \text{“Forgery is valid”}] \ge \frac{1}{X}.$$

By putting together both evaluations we obtain

$$\hat{\epsilon}_{\mathcal{A}} \geq \frac{\epsilon_{\mathcal{A}}}{t} \cdot \frac{|\tau_{(t-1,n,s-(n-1))}|}{|\tau_{(t,n,s)}|}$$

For the final part of the proof, we observe that distinguishing distributions $\mathcal{D}_C$ and $\mathcal{D}_{\textsc{Gen}}$ with $C'^{\mathcal{A}}$ is given by

$$\mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}) = |\hat{\epsilon}_A - \tilde{\epsilon}_{\mathcal{A}}|.$$

It also follows from $C'^{\mathcal{A}}$ that $\hat{\epsilon}_{\mathcal{A}}$ is an upper bound of $\tilde{\epsilon}_{\mathcal{A}}$. Hence, since $\hat{\epsilon}_{\mathcal{A}} \geq \tilde{\epsilon}_{\mathcal{A}}$, using the previous two equations, we get

$$\epsilon_{\mathcal{A}} \leq \frac{t \cdot |\tau_{(t,n,s)}|}{|\tau_{(t-1,n,s-(n-1))}|} \Big(\mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}) + \tilde{\epsilon}_{\mathcal{A}} \Big). \tag{6.6}$$

Now we employ the hybrid argument method to bound this advantage by the undetectability of $\mathcal{F}_\lambda$. Let

$$\mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}) = \sum_{\beta'=0}^{n-1} \Pr[\beta = \beta'] \cdot \mathrm{Adv}_{\mathcal{D}_C^{\beta=\beta'},\mathcal{D}_{\textsc{Gen}}^{\beta=\beta'}}(C'^{\mathcal{A}})$$

where $\mathcal{D}_C^{\beta=\beta'}$ and $\mathcal{D}_{\textsc{Gen}}^{\beta=\beta'}$ denote the distributions $\mathcal{D}_C$ and $\mathcal{D}_{\textsc{Gen}}$ with element β fixed to β'. This implies that there must exist at least one $\beta^\star$ such that

$$\mathrm{Adv}_{\mathcal{D}_C^{\beta=\beta^\star},\mathcal{D}_{\textsc{Gen}}^{\beta=\beta^\star}}(C'^{\mathcal{A}}) \geq \mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}).$$

Then, we define hybrids $H_i = \big(\beta^\star, c_k^{n-\beta^\star-i}(x, \mathbf{r}_{i+1:}), r, k\big)$ with $0 \leq i \leq n-\beta^\star$, r and k uniformly at random from their corresponding spaces and $x \leftarrow\!\$\, \{0,1\}^\lambda$. Hence, we can rewrite the previous inequality as

$$\mathrm{Adv}_{H_{n-\beta^\star},H_0}(C'^{\mathcal{A}}) \geq \mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}).$$

since $H_{n-\beta^\star}$ and H_0 coincide with $\mathcal{D}_C^{\beta=\beta^\star}$ and $\mathcal{D}_{\textsc{Gen}}^{\beta=\beta^\star}$, respectively. The hybrid argument states that there must exist two consecutive distributions $H_{i^\star}$ and $H_{i^\star+1}$ with $0 \leq i^\star < n-\beta^\star$ such that

$$\begin{aligned}\mathrm{Adv}_{H_{i^\star},H_{i^\star+1}}(C'^{\mathcal{A}}) \geq& \frac{1}{n-\beta^\star}\mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}})\\ \geq& \frac{1}{n}\mathrm{Adv}_{\mathcal{D}_C,\mathcal{D}_{\textsc{Gen}}}(C'^{\mathcal{A}}).\end{aligned}$$

Algorithm 7 $C''^{\mathcal{A}}$

```
Input: $\lambda$, sample $(u, k)$
Output: 0 or 1
Generate $\mathbf{r} \leftarrow\!\$\, \{0,1\}^{\lambda \times n}$
Call $C'^{\mathcal{A}}$ with inputs $\lambda$ and $\big(\beta^\star, c_k^{n-\beta^\star-(i^\star+1)}(u, \mathbf{r}_{i^\star+2:}), \mathbf{r}, k\big)$
return Output of $C'^{\mathcal{A}}$
```

Finally, assuming that $\mathcal{A}$ can distinguish H_0 and $H_{n-\beta^\star}$, we construct a machine $C''^{\mathcal{A}}$ that uses $C'^{\mathcal{A}}$ to break the undetectability of $\mathcal{F}_\lambda$, see Algorithm 7. Machine $C''^{\mathcal{A}}$ takes security parameter λ and a distinguishing challenge (u,k). Then, $C''^{\mathcal{A}}$ chooses $\mathbf{r}$ uniformly at random and calls $C'^{\mathcal{A}}$ with $\left(\beta^\star, c_k^{n-\beta^\star-(i^\star+1)}(u,\mathbf{r}_{i^\star+2:}),\mathbf{r},k\right)$ as input and returns its result. Denote a sample from distribution $\mathcal{D}_{\text{UD},\mathcal{U}}$ as taking $u \leftarrow\!\!\$\, \mathcal{U}_\lambda$ and $k \leftarrow\!\!\$\, \mathcal{K}_\lambda$ and a sample from $\mathcal{D}_{\text{UD},\mathcal{F}}$ as choosing $k \leftarrow\!\!\$\, \mathcal{K}_\lambda$ and $u \leftarrow f_k(\mathcal{U}_\lambda)$. Then, a sample (u,k) taken from $\mathcal{D}_{\text{UD},\mathcal{U}}$ entails that $c_k^{n-\beta^\star-(i^\star+1)}(\mathcal{U}_\lambda,\mathbf{r}_{i^\star+2:})$ is distributed exactly like the second element of the tuple $H_{i^\star+1}$. Furthermore, a sample (u,k) from $\mathcal{D}_{\text{UD},\mathcal{F}}$ implies that

$$\begin{aligned}
&c_k^{n-\beta^\star-(i^\star+1)}(f_k(\mathcal{U}_\lambda),\mathbf{r}_{i^\star+2:})\\
&= c_k^{n-\beta^\star-(i^\star+1)+1}(\mathcal{U}_\lambda \oplus \mathbf{r}_{i^\star+1},\mathbf{r}_{i^\star+1:})\\
&= c_k^{n-\beta^\star-i^\star}(\mathcal{U}_\lambda,\mathbf{r}_{i^\star+1:})
\end{aligned}$$

is distributed exactly like the second element of the tuple $H_{i^\star}$, where we use the fact that the $\oplus$ operator of a uniformly distributed value with some other value with arbitrary distribution results in a uniformly distributed value. From this we have that

$$\text{Adv}_{\mathcal{D}_{\text{UD},\mathcal{U}},\mathcal{D}_{\text{UD},\mathcal{F}}}(C''^{\mathcal{A}}) = \text{Adv}_{H_{i^\star},H_{i^\star+1}}(C'^{\mathcal{A}})$$

and furthermore

$$\text{Adv}_{\mathcal{D}_{\text{UD},\mathcal{U}},\mathcal{D}_{\text{UD},\mathcal{F}}}(C''^{\mathcal{A}}) \leq \text{InSec}^{\text{UD}}(\mathcal{F}_\lambda;\tilde{z})$$

with bounded running time $\tilde{z} = z+3tn+n$. This bound is obtained from the running time cost of running $C'^{\mathcal{A}}$ that is the same as the previously calculated running time $z' = z+3tn$, in addition to at most $n-1$ calls to f_k for $C''^{\mathcal{A}}$ to initialize the input before calling $C'^{\mathcal{A}}$. Lastly, by joining these last results, we have

$$\text{Adv}_{\mathcal{D}_C,\mathcal{D}_{\text{GEN}}}(C'^{\mathcal{A}}) \leq n\cdot\text{InSec}^{\text{UD}}(\mathcal{F}_\lambda;\tilde{z}).$$

that yields

$$\epsilon_{\mathcal{A}} \leq \frac{t\cdot|\tau_{(t,n,s)}|}{|\tau_{(t-1,n,s-(n-1))}|}\left(n\cdot\text{InSec}^{\text{UD}}(\mathcal{F}_\lambda;\tilde{z})+\tilde{\epsilon}_{\mathcal{A}}\right)$$

by using (6.6). Finally, by applying (6.5) we obtain the bound on $\epsilon_{\mathcal{A}}$ that leads to the contradiction. □

6.5.3 Security level

We now follow the same argument as in (HÜLSING, 2013b; KUDINOV; KIKTENKO; FEDOROV, 2020) to compute the security level of our variant signature scheme, as previously defined in Section 2.2. WOTS-CS+ has a security level q if a successful attack is expected to undertake, on average, 2^{q-1} evaluations of the one-way function from the family $\mathcal{F}_\lambda$. Hence, we derive a lower bound for z by considering

$$\text{InSec}^{\text{EU-CMA}}(\text{WOTS-CS+}(\lambda,\ell,t,n,s);z,1) \geq \frac{1}{2}.$$

We now use the same assumptions as related works (HÜLSING, 2013b; KUDINOV; KIKTENKO; FEDOROV, 2020) regarding the insecurity of $\mathcal{F}_\lambda$, following that

$$\mathrm{InSec}^{\mathrm{UD}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{OW}}(\mathcal{F}_\lambda; z) = \mathrm{InSec}^{\mathrm{SPR}}(\mathcal{F}_\lambda; z) = \frac{z}{2^\lambda}.$$

Furthermore, since $z \gg 4tn$ for practical attacks, we can use the fact that $z \approx \tilde{z} \approx z'$ with negligible difference. We solve the security level bound for z as

$$\frac{t \cdot |\tau_{(t,n,s)}|}{|\tau_{(t-1,n,s-(n-1))}|}\left(n\frac{z}{2^\lambda} + \frac{z}{2^\lambda} + n\frac{z}{2^\lambda}\right) \geq \frac{1}{2}$$
$$\frac{t \cdot |\tau_{(t,n,s)}|}{|\tau_{(t-1,n,s-(n-1))}|}(2n+1)\frac{z}{2^\lambda} \geq \frac{1}{2}.$$

Finally, we have that

$$z \geq 2^\lambda \cdot \frac{1}{2} \cdot \frac{1}{(2n+1)} \cdot \frac{|\tau_{(t-1,n,s-(n-1))}|}{t \cdot |\tau_{(t,n,s)}|}$$
$$z \geq 2^{\lambda - 1 + \log_2(|\tau_{(t-1,n,s-(n-1))}|) - \log_2(t(2n+1)|\tau_{(t,n,s)}|)}$$

yields the security bound

$$q \geq 2^{\lambda + \log_2(|\tau_{(t-1,n,s-(n-1))}|) - \log_2(t(2n+1)|\tau_{(t,n,s)}|)}$$

against classical adversaries and

$$q \geq 2^{\frac{\lambda}{2} + \log_2(|\tau_{(t-1,n,s-(n-1))}|) - \log_2(t(2n+1)|\tau_{(t,n,s)}|)}$$

against quantum adversaries.

Table 15 – Security level q for WOTS+ and WOTS-CS+ and $\lambda = \ell = 256$.

Adversary	t	w	n	s	WOTS+	WOTS-CS+
Classical	34	256	226	3643	233.91	233.99
	67	16	15	400	240.89	240.19
Quantum	34	256	226	3643	105.91	105.99
	67	16	15	400	112.89	112.19

Source: The author.

To give a better idea of the security level, we compare our bound to the ones given in (KUDINOV; KIKTENKO; FEDOROV, 2020) for classical and quantum attacks. We observe that we consider $\frac{\lambda}{2}$ for quantum attackers using Grover's algorithm, as also presented in previous works. The parameters in Table 15 match those of Table 4 and Table 5 with $\ell = 256$ for the respective value of t. The WOTS+ security level against classical and quantum attacks, according to (KUDINOV; KIKTENKO; FEDOROV, 2020), are given respectively by

$$q > \lambda - \log_2(tw) - \log_2(2w+1),$$
$$q > \frac{\lambda}{2} - \log_2(tw) - \log_2(2w+1).$$

6.6 FURTHER DEVELOPMENTS

Algorithm 8 Deterministic constant-sum encoding DCS for alternate parameters

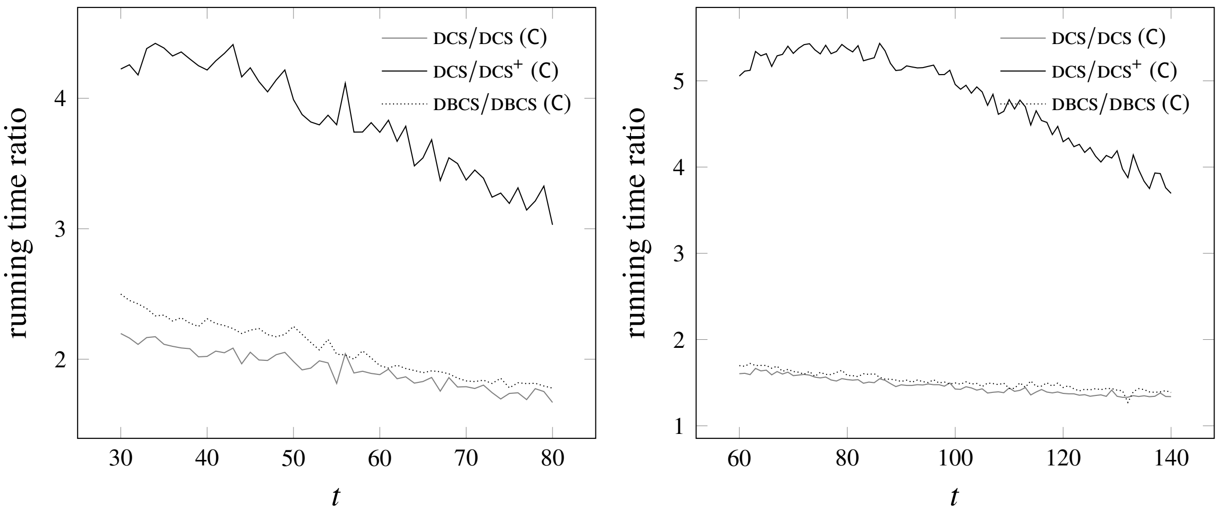
```
Input: $t, n, s \in \mathbb{N}$, $I \in \iota_{(t,n,s)}$
Output: $(b_1, \dots, b_t) \in \tau_{(t,n,s)}$
if $t = 1$ then
    return $(s)$
$b \leftarrow 0$
$h_l \leftarrow 0$
$h_r \leftarrow |\tau_{(t-1,n,s)}|$
while not $h_l \le I < h_r$ do
    $b \leftarrow b + 1$
    $h_l \leftarrow h_r$
    $h_r \leftarrow h_r + |\tau_{(t-1,n,s-b)}|$
return $(n - b)$ || DCS$(t-1, n, s-b, I-h_l)$
```

During the submission process of (PERIN et al., 2021), it was noted that there are application scenarios for the alternative parameters, suggested in Section 6.4.2. The results for these parameters are given without considering the running time of the encoding algorithm, justified by the application scenario. However, it turns out that the running times of the DCS and DBCS may take a huge hit in performance, due to the increased value s of the alternate parameters. This motivated us to include a slight modification to Algorithm 2 that completely avoids this performance issue. Recall from Proposition 6.1.1 that $|\tau_{(t,n,s)}| = |\tau_{(t,n,tn-s)}|$ and if $(x_1, \dots, x_t) \in \tau_{(t,n,s)}$, then we must have that $(n - x_1, \dots, n - x_t) \in \tau_{(t,n,tn-s)}$. We restate DCS in Algorithm 8 with the modification highlighted in red for completion, and remark that the same change can be trivially implemented for DBCS.

Another contribution that was not included in the constant-sum paper, was later investigated by (ZAMBONIN, 2021). Two key observations that are made in the report: the first is that the particular case of the encoding benchmarks of Figure 10 is implemented in C++ and uses techniques such as object oriented and templates to instantiate each parameter set of the test. By using the C implementation, similar to the one used for the XMSS experiment, it is possible to remove the overhead caused by these techniques and improve the performance of the encoding function by up to 2.19 and 2.49 for DCS and DBCS respectively, when $m = 256$. Similar performance gains are also reported for $m = 512$.

The main contribution of (ZAMBONIN, 2021) is the improved DCS+. This modified algorithm reuses values that are computed in each iteration of the DCS algorithm, to avoid heavy and repeated computations of large binomials. We show in Figure 11 the ratio of the running time of DCS and DBCS divided by the respective implementation in C. The figure also shows the ratio of the running time of DCS divided by DCS$^+$. Considering the latter and for parameters $t = 34$ and $\ell = 256$, the encoding DCS$^+$ algorithm outperforms our implementation running 4.42 times faster. Moreover, the author also provide further investigations of memory consumption of the recursive algorithms, compared to an iterative version.

Figure 11 – Running time ratio of DCS and DBCS implemented in C++ divided by the respective implementation in C and DCS$^+$.



(a) $\ell = 256$ and $30 \leq t \leq 80$.

(b) $\ell = 512$ and $60 \leq t \leq 140$.

Source: Zambonin (2021)

It is interesting to observe, despite the achievements in (ZAMBONIN, 2021), that the discussion of the performance of the constant-sum encoding algorithms in 6.4 still holds. Unfortunately, it seems that for the general applicability of our algorithms one would have to reduce running time by at least two or three orders of magnitude.

7 FINAL REMARKS

In this chapter, we discuss the results we have achieved with our contributions. As we have discussed on each respective chapter, there are several improvements that we have found for each of our proposals. However, there are implications regarding trade-off costs or increased complexity to the encoding algorithm that prevent our techniques to be considered a general purpose drop in solution. For this reason, we provide a summary of scenarios where we think we can implement our proposals successfully. We finalize with an overview of our accomplishments and relevant open problems that we leave for future work.

7.1 APPLICATIONS

7.1.1 Relevant application cases for WOTS-BR

As it was originally intended, WOTS-BR was proposed as a general case solution for digital signatures. This is mainly due to the particular case where we expect to verify documents many times, but the signature occurs only once. We show that with a few iterations, i.e. $R = 25$, we can already have faster verification speeds in average. As we increase R, this becomes more noticeable, but signature costs thus are increased. Unfortunately, for the case of improving signature speeds, there is only a few scenarios where WOTS-R can produce positive results. This is mainly due to the fact that the R iterations have to be performed by the signer. Hence, we believe that WOTS-CS+ performs better in this scenario. We give more details on this in the next section.

The main case where WOTS-BR might play its part today was pointed out in (BOS et al., 2020). As an example, the authors mention firmware updates and secure boot for IoT devices. In such cases, signatures can be generated using high-end and powerful processors, much more efficiently. The verification is performed several times, by devices with limited computational capabilities. Hence, WOTS-BR can be successfully deployed in such cases to reduce the cost for these devices to assert the integrity and authenticity of a firmware update. Indeed, if time is also not a constraint, authors in (BOS et al., 2020, Table 7.2) give performance estimates when committing to minimize signature verification costs with WOTS-R for minutes, hours, days and even a week.

7.1.2 Relevant application cases for WOTS-CS+

The main applications that can benefit from our contributions with WOTS-CS+ are those that require predictability of the costs associated with the generation of keys and signatures, and the subsequent signature verification. These applications are most commonly found on computing devices with limited resources. However, the benefit of predictability leads to an increased encoding cost. Therefore, through examples, we present applications where the encoding costs can be adequately treated, clarifying the benefits of this alternative.

A case of interest is a typical scenario in which devices such as smart cards are used to sign electronic documents. In general, these devices generate private keys with minimal memory and processing capabilities, without ever exposing them to the outside world. For a document to be signed, it is necessary to determine its hash externally to the smart card. In practice, this step is done by computers with higher computational power, and the hash is then sent to the smart card to be signed. Upon receiving the signature, the external computer proceeds with the assembly of the representation structure of the signed document (RANKL; EFFING, 2010, Chapter 23). Considering this scenario, the constant-sum encoding can be previously computed and sent to the signing device. We can also choose the exact number of hash operations that we want to perform during the signature by observing the parameters proposed in this work. We show in Section 6.4.2 that we can improve the generation of signatures for this case, using parameters compatible to the standard ones for WOTS+ (HÜLSING et al., 2018).

The second scenario of considerable practical importance is the preservation of the long-term authenticity of data. For example, consider large signature databases that need to be updated to be resistant against quantum attacks. Such applications might require a substantial volume of signatures to be generated over time. Since hash-based signatures are most likely going to be deployed in secure platforms that can guarantee state integrity (COOPER et al., 2020), it is also likely that the signing device will be provided only with the necessary data to perform the signature, despite its CPU capabilities. Hence, a trusted party can perform the document digest and its encoding, and place it in a queue. In this case, the signing device may benefit from our parameter selection, with a smaller and fixed number of hash operations to perform for each signature, increasing its throughput. Furthermore, to decrease storage space requirements, less traditional parameters can be used to output WOTS-CS+ signatures that are close to half of the size required by using standard parameters (e.g., $t = 34$ in Table 11). We also obtain significant improvements to key generation costs for these new parameter selections, which may also impact signature performance, further decreasing the number of required hash operations.

Finally, in some other situations, it is interesting to focus on verifying signatures more efficiently, as discussed in the case of WOTS-R. Considering this scenario, WOTS-CS+ can control and reduce the number of hash operations needed to be performed in the verification process. This case is more challenging to implement, as we have seen that the costs related to the encoding may substantially diminish the gains with the reduced verification cost, as well as being impractical for some devices. We recall that, whenever possible, the encoding can be included in the signed message, which the verifier can either trust or use to reduce the encoding cost with VCS, before verifying the signature.

7.2 ACHIEVEMENTS AND CONCLUSION

In this thesis, we have reviewed encoding alternatives that were proposed to replace the `base-w` encoding originally in Wots and its state of the art variant Wots+. We have proposed two major contributions, namely Wots-br and Wots-cs+. The first is a composition of two simple techniques that produce remarkable results for improving signature verification running time with a trade-off where signature generation running time is increased. The second is an improvement of the constant-sum encoding alternative, that minimizes the cost of key generation for Winternitz-based schemes, thus reducing signing and verifying running times as well.

For Wots-br, we have introduced a padding to the `base-w` encoding that enables faster verification for Wots+ and consequently Xmss. This proposal is not compatible with RFC 8391, but we hope it is considered for future revisions of hash-based signature standards. Furthermore, we have proposed a technique to randomize the message to be signed that can greatly increase verification performance. This technique was later adapted to be compatible with RFC 8391 and shown to reduce the average verification cost of a single Wots+ signature by up to 55.5% (BOS et al., 2020, Table 7.2). The same authors use our contribution, along with other modifications proposed in the literature, to achieve Xmss verification running time in Cortex-M4 devices at half the expected average time.

For the case of Wots-cs+, we introduce new deterministic encoding algorithms for the constant-sum variant of the Winternitz one-time signature scheme, Wots-cs. Our methods have reduced costs associated with the encoding function by employing distinct techniques such as dynamic programming and binary search. Not only do we achieve faster encoding than the probabilistic alternative in the literature, but we expand the sets of parameters that can be used with Wots-cs, by aiming to reduce the cost of key generation for hash-based signatures.

As a result, our approach allows for a flexible trade-off between costs of key generation, signature generation and signature verification, by accepting both strategies for selecting parameters: MinVer or MinGen. Our approach allows us to reduce the number of applications of the chaining function for the key generation step, achieving better costs that are more competitive to those obtained with Wots+. We emphasize that reducing the cost of key generation is particularly interesting, since it decreases the overall cost for all signature-related steps and is not covered by other constant-sum-related works (CRUZ; YATANI; KAJI, 2016; KAJI; CRUZ; YATANI, 2018).

For specific sets of parameters, our parameter selection can yield faster key generation, signature generation and signature verification simultaneously, when compared to Wots+; see Table 11. Indeed, these results are reflected in other hash-based signature schemes, such as Xmss; see $t = 34$ in Table 12. We also observe that the reduction factor in the key generation cost is invariant to the height of the Xmss tree, that is, the number of leaves containing a Winternitz scheme instance. Experiments in Table 12 show the potential improvements of Wots-cs+ for all sets of parameters, if better encoding algorithms are discovered.

Alternatively, we propose distinct parameters that can be used with specific applications, reducing the cost of signature generation instead of verification. This is a relevant scenario for many cases related to the signature of electronic documents. We argue that signing devices can take advantage of the predictable and reduced cost for signing without having to deal with the encoding cost (Section 6.4.2).

Another significant contribution of our work is the comprehensive study of the constant-sum tuples. We prove several interesting and fundamental properties that are crucial for the security analysis of the scheme. Under a wide range of acceptable parameters, we prove that Wots-cs+ is EU-CMA provided that $\mathcal{F}_\lambda$ is a one-way, second preimage resistant and undetectable function family. This is a major improvement over Wots-cs (KAJI; CRUZ; YATANI, 2018), placing Wots-cs+ on pair with the security of Wots+ for the same signature sizes. Previously, the constant-sum scheme security relied on stronger assumptions such as collision resistance.

7.2.1 Open problems and future work

At the end of Section 6.4.2, we have indicated that alternative parameter combinations that reduce Sig in the context of Xmss could be tailored for specific applications. Such studies are an interesting avenue for further research.

While we believe that several applications may benefit from our contributions, we acknowledge that the encoding performance of Wots-cs+ needs to be improved for widespread applicability. This obstacle could be overcome by future research on more efficient encoding algorithms for the constant-sum tuples. It may be possible to adapt combinatorial algorithms that apply to classic problems, such as unranking a generalized composition of a number (KNUTH, 2011). The constant-sum encoding functions studied so far use a lexicographical order of tuples in $\mathcal{T}_{(t,n,s)}$. Future work could improve encoding performance for this ordering or perhaps they could use a different ordering of tuples in $\mathcal{T}_{(t,n,s)}$ that may lead to more efficient encoding algorithms. We hope that this could potentially make Wots-cs+ more widely applicable.

Lastly, we also believe that by reducing costs related to some part of the signature algorithms might also produce interesting power consumption studies. For example, it might be interesting to understand better how Wots-r power consumption is affected as R is increased, when verifying signature with IoT devices.

BIBLIOGRAPHY

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

AUMASSON, J.-P. et al. **SPHINCS$^+$**. 2020. Submission to the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Disponível em: https://sphincs.org/data/sphincs+-round3-submission-nist.zip.

BERNSTEIN, D. J.; BUCHMANN, J.; DAHMEN, E. **Post Quantum Cryptography**. 1st. ed. [S.l.]: Springer, 2008. ISBN 3540887016.

BERNSTEIN, D. J. et al. The SPHINCS$^+$ Signature Framework. In: **Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security**. [S.l.: s.n.], 2019.

BOLLINGER, R. C.; BURCHARD, C. L. Lucas's Theorem and Some Related Results for Extended Pascal Triangles. **The American Mathematical Monthly**, v. 97, n. 3, p. 198–204, mar. 1990.

BONEH, D.; SHEN, E.; WATERS, B. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In: YUNG, M. et al. (Ed.). **Public Key Cryptography - PKC 2006**. [S.l.: s.n.], 2006. (Lecture Notes in Computer Science, v. 3958), p. 229–240.

BONEH, D.; SHOUP, V. **A graduate course in applied cryptography**. [S.l.]: Self-publishing, 2020.

BOS, J. W. et al. Rapidly Verifiable XMSS Signatures. **IACR Transactions on Cryptographic Hardware and Embedded Systems**, v. 2021, n. 1, p. 137–168, dez. 2020.

BRUINDERINK, L. G.; HÜLSING, A. "Oops, I did it again"–Security of One-Time Signatures under Two-Message Attacks. In: SPRINGER. **International Conference on Selected Areas in Cryptography**. [S.l.], 2017. p. 299–322.

BUCHMANN, J. et al. On the security of the Winternitz one-time signature scheme. In: SPRINGER. **International Conference on Cryptology in Africa**. [S.l.], 2011. p. 363–378.

BUCHMANN, J.; DAHMEN, E.; SCHNEIDER, M. Merkle Tree Traversal Revisited. In: BUCHMANN, J.; DING, J. (Ed.). **Post-Quantum Cryptography**. [S.l.: s.n.], 2008. (Lecture Notes in Computer Science, v. 5299), p. 63–78.

COOPER, D. A. et al. **Recommendation for Stateful Hash-Based Signature Schemes**. [S.l.], 2020.

CRUZ, J. P.; YATANI, Y.; KAJI, Y. Constant-Sum Fingerprinting for Winternitz One-Time Signature. In: **2016 International Symposium on Information Theory and Its Applications (ISITA)**. [S.l.: s.n.], 2016. p. 703–707.

DIFFIE, W.; HELLMAN, M. New directions in cryptography. **IEEE transactions on Information Theory**, IEEE, v. 22, n. 6, p. 644–654, 1976.

DODS, C.; SMART, N. P.; STAM, M. Hash based digital signature schemes. In: SPRINGER. **IMA International Conference on Cryptography and Coding**. [S.l.], 2005. p. 96–115.

EGER, S. Stirling's Approximation for Central Extended Binomial Coefficients. **The American Mathematical Monthly**, v. 121, n. 4, p. 344–349, dez. 2014.

ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. **IEEE transactions on information theory**, IEEE, v. 31, n. 4, p. 469–472, 1985.

FAHSSI, N.-E. **Polynomial Triangles Revisited**. 2012. Disponível em: https://arxiv.org/abs/1202.0228v7.

GHEORGHIU, V.; MOSCA, M. A Resource Estimation Framework For Quantum Attacks Against Cryptographic Functions: Recent Developments. 2021.

GOLDWASSER, S.; BELLARE, M. **Lecture Notes on Cryptography**. 2008. Disponível em: https://cseweb.ucsd.edu/~mihir/papers/gb.pdf.

GOLOMB, S. W.; GONG, G. **Signal design for good correlation: for wireless communication, cryptography, and radar**. [S.l.]: Cambridge University Press, 2005.

GROVER, L. K. A fast quantum mechanical algorithm for database search. In: **Proceedings of the twenty-eighth annual ACM symposium on Theory of computing**. [S.l.: s.n.], 1996. p. 212–219.

HÅSTAD, J. et al. A pseudorandom generator from any one-way function. **SIAM Journal on Computing**, SIAM, v. 28, n. 4, p. 1364–1396, 1999.

HELLESETH, T. Golomb's randomness postulates. In: ____. **Encyclopedia of Cryptography and Security**. Boston, MA: Springer US, 2011. p. 516–517. ISBN 978-1-4419-5906-5. Disponível em: https://doi.org/10.1007/978-1-4419-5906-5_351.

HÜLSING, A. Practical forward secure signatures using minimal security assumptions. Technische Universität, 2013.

HÜLSING, A. W-OTS$^+$ – Shorter Signatures for Hash-Based Signature Schemes. In: YOUSSEF, A.; NITAJ, A.; HASSANIEN, A. E. (Ed.). **Progress in Cryptology – AFRICACRYPT 2013**. [S.l.: s.n.], 2013. (Lecture Notes in Computer Science, v. 7918), p. 173–188.

HÜLSING, A. et al. **XMSS: Extended Hash-Based Signatures**. [S.l.], 2018. Disponível em: https://tools.ietf.org/html/rfc8391.

KAJI, Y.; CRUZ, J. P.; YATANI, Y. Hash-Based Signature with Constant-Sum Fingerprinting and Partial Construction of Hash Chains. In: **Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, ICETE 2018 - Volume 2: SECRYPT**. [S.l.: s.n.], 2018. p. 463–470.

KNUTH, D. E. **The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2011. ISBN 0201038048.

KUDINOV, M. A.; KIKTENKO, E. O.; FEDOROV, A. K. **Security analysis of the W-OTS$^+$ signature scheme: Updating security bounds**. 2020. Disponível em: https://arxiv.org/abs/2002.07419v1.

LAMPORT, L. **Constructing digital signatures from a one-way function**. [S.l.], 1979.

LENSTRA, A. K. Key length. Contribution to the handbook of information security. Citeseer, 2004.

LEURENT, G.; PEYRIN, T. **SHA-1 is a Shambles - First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust**. 2020. Cryptology ePrint Archive, Report 2020/014. Disponível em: https://eprint.iacr.org/2020/014.

MCGREW, D.; CURCIO, M.; FLUHRER, S. **Leighton-Micali hash-based signatures**. [S.l.], 2019. Disponível em: https://tools.ietf.org/html/rfc8554.

MERKLE, R. C. A Certified Digital Signature. In: BRASSARD, G. (Ed.). **Advances in Cryptology – CRYPTO '89**. [S.l.: s.n.], 1989. (Lecture Notes in Computer Science, v. 435), p. 218–238.

NIEHUES, L. B. et al. Sidon sets and statistics of the ElGamal function. **Cryptologia**, Taylor & Francis, v. 44, n. 5, p. 438–450, 2020.

NIST. **Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process**. [S.l.], 2016. Disponível em: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals.

OKEYA, K. et al. Signed binary representations revisited. In: SPRINGER. **Annual International Cryptology Conference**. [S.l.], 2004. p. 123–139.

PANARIO, D.; PERIN, L. P.; STEVENS, B. Comparing balanced $\mathbb{Z}_v$-sequences obtained from ElGamal function to random balanced sequences. To be submitted. N.D.

PERIN, L. P. et al. Improved constant-sum encodings for hash-based signatures. **Journal of Cryptographic Engineering**, Springer, p. 1–23, 2021.

PERIN, L. P. et al. Tuning the Winternitz Hash-Based Digital Signature Scheme. In: **2018 IEEE Symposium on Computers and Communications (ISCC)**. [S.l.: s.n.], 2018. p. 537–542.

PRODINGER, H. On binary representations of integers with digits- 1, 0, 1. **Integers**, p. A8–14, 2000.

RANKL, W.; EFFING, W. **Smart Card Handbook**. 4th. ed. [S.l.]: Wiley, 2010. ISBN 0470743670.

ROGAWAY, P.; SHRIMPTON, T. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: SPRINGER. **International workshop on fast software encryption**. [S.l.], 2004. p. 371–388.

ROH, D.; JUNG, S.; KWON, D. Winternitz Signature Scheme Using Nonadjacent Forms. **Security and Communication Networks**, v. 2018, jun. 2018.

ROMPEL, J. One-way functions are necessary and sufficient for secure signatures. In: **Proceedings of the twenty-second annual ACM symposium on Theory of computing**. [S.l.: s.n.], 1990. p. 387–394.

STEINWANDT, R.; VILLÁNYI, V. I. A one-time signature using run-length encoding. **Information Processing Letters**, v. 108, n. 4, p. 179–185, out. 2008.

STEVENS, M. et al. The first collision for full SHA-1. In: KATZ, J.; SHACHAM, H. (Ed.). **Advances in Cryptology – CRYPTO 2017**. [s.n.], 2017. (LNCS, v. 10401), p. 570–596. Disponível em: https://eprint.iacr.org/2017/190.

ZAMBONIN, G. Restricted integer compositions in the Winternitz signature scheme. Unpublished report. 2021.