

Practical algorithms and parameters for modification-tolerant signature scheme

Anthony Bernardo Kamers¹ Paola de Oliveira Abel¹ Thaís Bardini
Idalino¹ Gustavo Zambonin¹ Jean Everson Martina¹

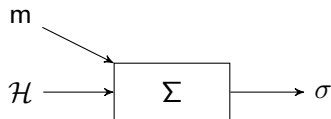
¹Universidade Federal de Santa Catarina - UFSC

17 de Setembro de 2024

Traditional digital signatures

What happens if we change the signed document?

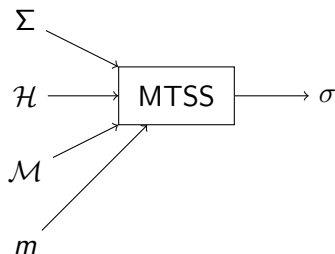
Traditional digital signatures



- $\mathcal{H}$ is a hash function
- Σ is a traditional signature scheme
- σ is the signature

MTSS

Modification-tolerant signature scheme [1]



- $\mathcal{M}$ is a table with special properties

[1] Thaís Bardini Idalino, Lucia Moura, and Carlisle Adams. “Modification tolerant signature schemes: location and correction”. In: *International Conference on Cryptology in India*. Springer. 2019, pp. 23–44.

Cover-free families (CFFs)

2-CFF(9, 12)

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
1	1	0	0	1	0	0	1	0	0	1	0	0
2	1	0	0	0	1	0	0	1	0	0	1	0
3	1	0	0	0	0	1	0	0	1	0	0	1
4	0	1	0	1	0	0	0	0	1	0	1	0
5	0	1	0	0	1	0	1	0	0	0	0	1
6	0	1	0	0	0	1	0	1	0	1	0	0
7	0	0	1	1	0	0	0	1	0	0	0	1
8	0	0	1	0	1	0	0	0	1	1	0	0
9	0	0	1	0	0	1	1	0	0	0	1	0

Cover-free families (CFFs)

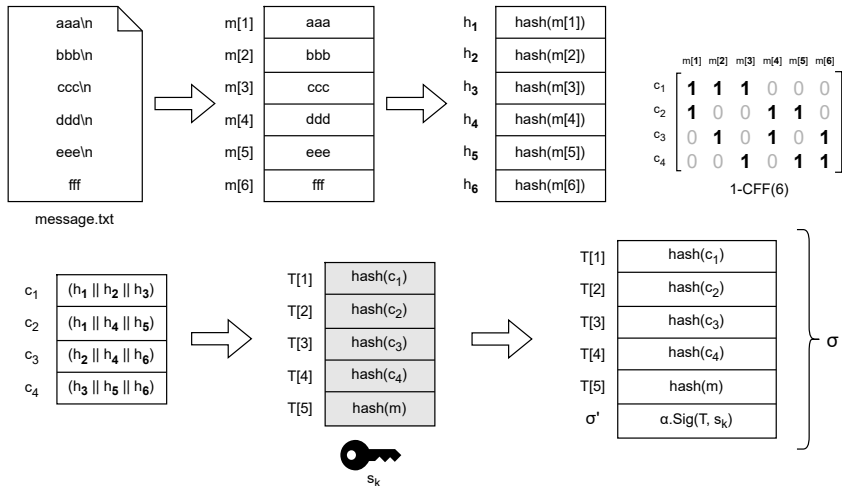
Identifying defects using CFFs

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
Test 1 ✗	1	0	0	1	0	0	1	0	0	1	0	0
Test 2 ✗	1	0	0	0	1	0	0	1	0	0	1	0
Test 3 ✓	1	0	0	0	0	1	0	0	1	0	0	1
Test 4 ✗	0	1	0	1	0	0	0	0	1	0	1	0
Test 5 ✓	0	1	0	0	1	0	1	0	0	0	0	1
Test 6 ✗	0	1	0	0	0	1	0	1	0	1	0	0
Test 7 ✗	0	0	1	1	0	0	0	1	0	0	0	1
Test 8 ✓	0	0	1	0	1	0	0	0	1	1	0	0
Test 9 ✓	0	0	1	0	0	1	1	0	0	0	1	0

$d\text{-CFF}(t, n)$

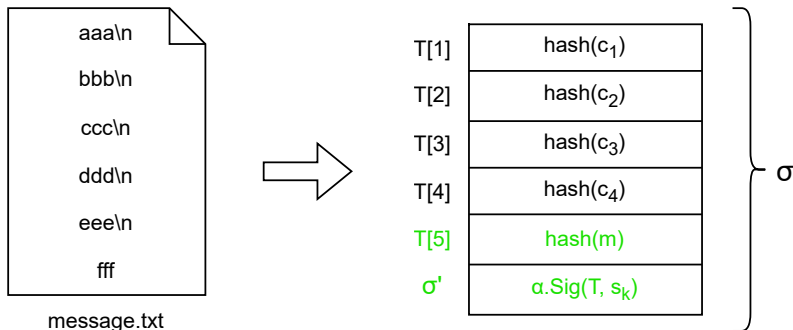
MTSS

Signature process - Algorithm SIG



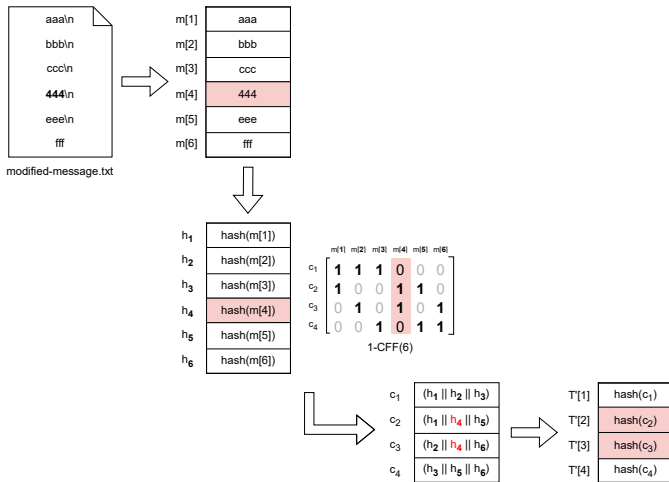
MTSS

Verifying process - Algorithm V_{ER}

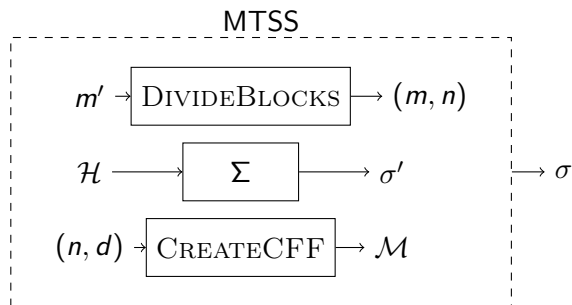


MTSS

Locating process - Algorithm VER

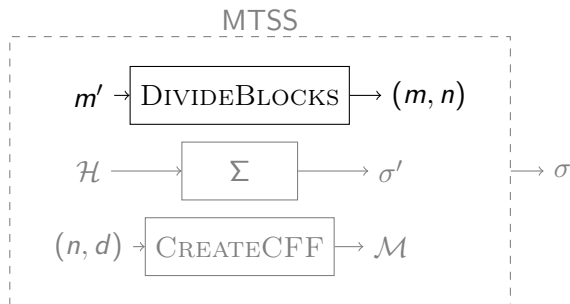


- Complementing the MTSS framework [1]
 - High-level implementation
 - Performance statistics
 - CFF parameters
 - How to efficiently divide a document into blocks
- New usage for MTSS: provide integrity and authenticity of any part of a signed document without ownership of the whole message



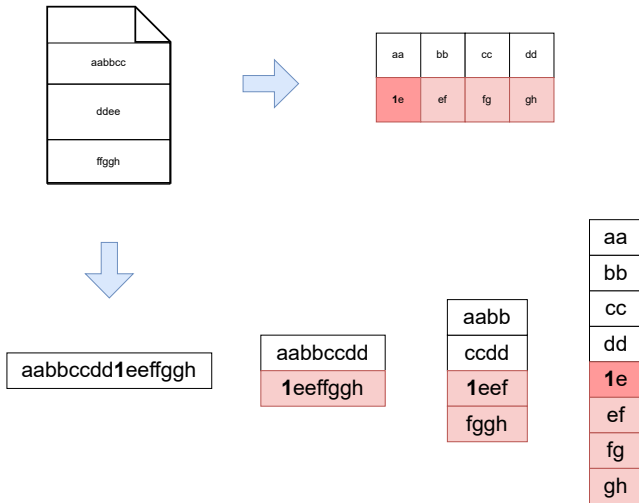
MTSS in practice

Dividing blocks



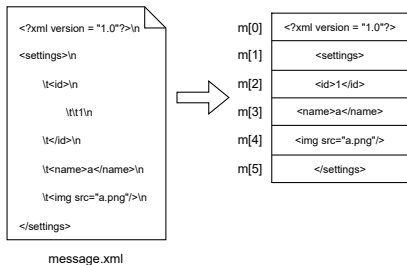
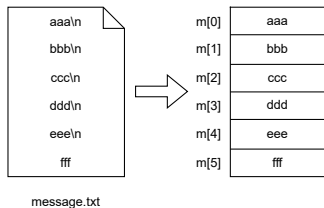
MTSS in practice

Dividing blocks: approaches



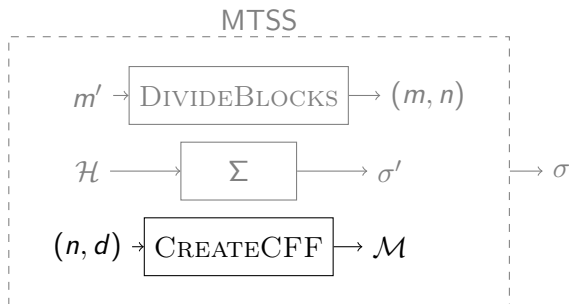
MTSS in practice

Dividing blocks: our approach



MTSS in practice

Creating CFFs

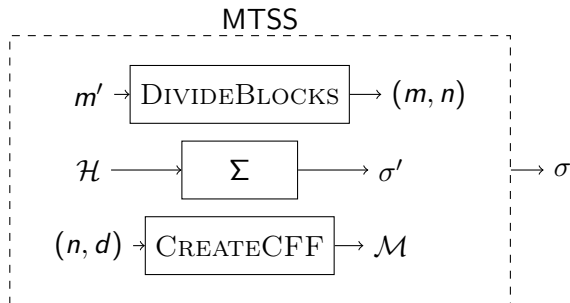


Result: poor performance generation

Solution: cache!!

MTSS in practice

Experiments using different Σ and $\mathcal{H}$



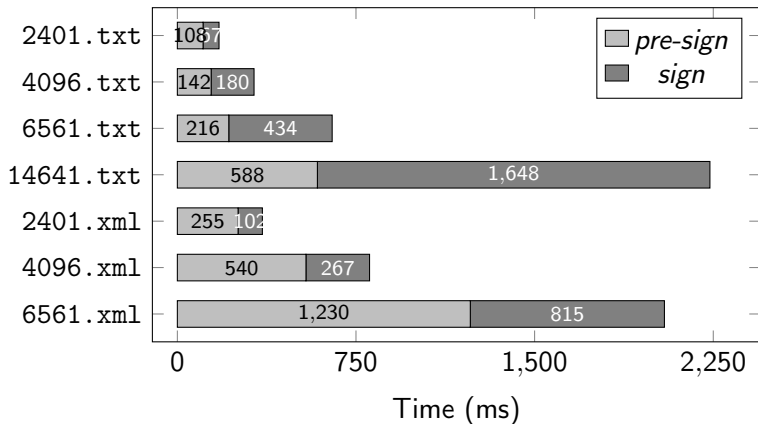
MTSS in practice

SIG algorithm using $\mathcal{M} = 2\text{-CFF}(25, 125)$

		SIG time (ms)						$ \sigma $ (bytes)	
		SHA-2		SHA-3		BLAKE			
Σ		256	512	256	512	2s	2b	256	512
Raw Σ	RSA-2048	4.83	3.63	3.93	6.42	2.49	3.32	256	256
	ML-DSA-44	3.93	2.67	2.96	5.49	1.54	2.36	2360	2360
	Ed25519		3.08						64
MTSS	RSA-2048	27.35	19.42	21.76	36.85	10.86	15.63	1088	1880
	ML-DSA-44	26.44	18.85	21.27	36.04	10.36	15.2	3180	3990
	Ed25519		19.99						1690

MTSS in practice

SIG with `DIVIDEBLOCKS` and `CREATECFF`



MTSS in practice

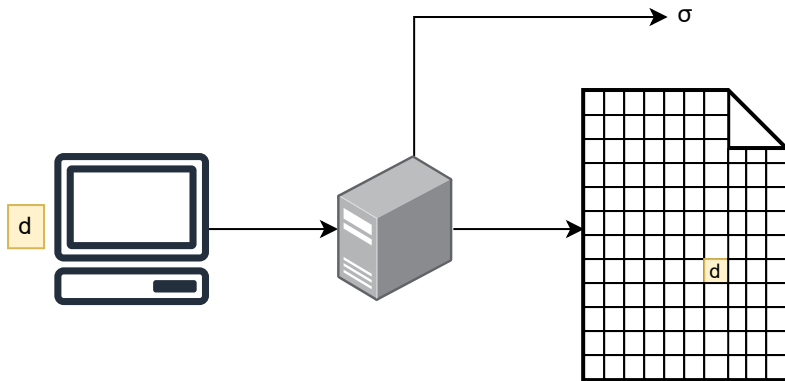
V_{ER} algorithm using $\mathcal{M} = 2\text{-CFF}(25, 125)$

		V_{ER} time (ms)					
		Σ					
		SHA-2		SHA-3		BLAKE	
		256	512	256	512	2s	2b
Raw Σ	RSA-2048	4.09	2.87	3.15	5.66	3.21	4.84
	ML-DSA-44	3.88	2.61	2.90	5.47	1.51	2.30
	Ed25519		3.84				
$ V = 0$	RSA-2048	4.16	2.95	3.25	5.80	1.81	2.64
	ML-DSA-44	3.94	2.69	3.00	5.55	1.55	2.37
	Ed25519		3.90				
$ V = 1$	RSA-2048	159.95	174.53	162.22	162.29	153.15	162.94
	ML-DSA-44	156.93	177.55	159.49	165.35	163.51	154.37
	Ed25519		165.41				

What if we did not need to sign every page of the Brazilian Federal Register (Diário Oficial da União) to verify one page's integrity and authenticity?

Ensuring data integrity for individual blocks

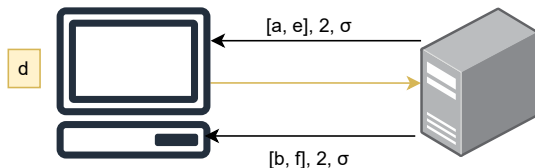
Using MTSS



Ensuring data integrity for individual blocks

Using MTSS

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
1	1	1	1	0	0	0
2	1	0	0	1	1	0
3	0	1	0	1	0	1
4	0	0	1	0	1	1



Practical algorithms and parameters for modification-tolerant signature scheme

- Repository: <https://github.com/AnthonyKamers/mtss-signer>

Anthony Kamers	<code>anthony.kamers@posgrad.ufsc.br</code>
Paola Abel	<code>paola.abel@grad.ufsc.br</code>
Thaís Bardini	<code>thais.bardini@ufsc.br</code>
Gustavo Zambonin	<code>gustavo.zambonin@posgrad.ufsc.br</code>
Jean Martina	<code>jean.martina@ufsc.br</code>

Questions?