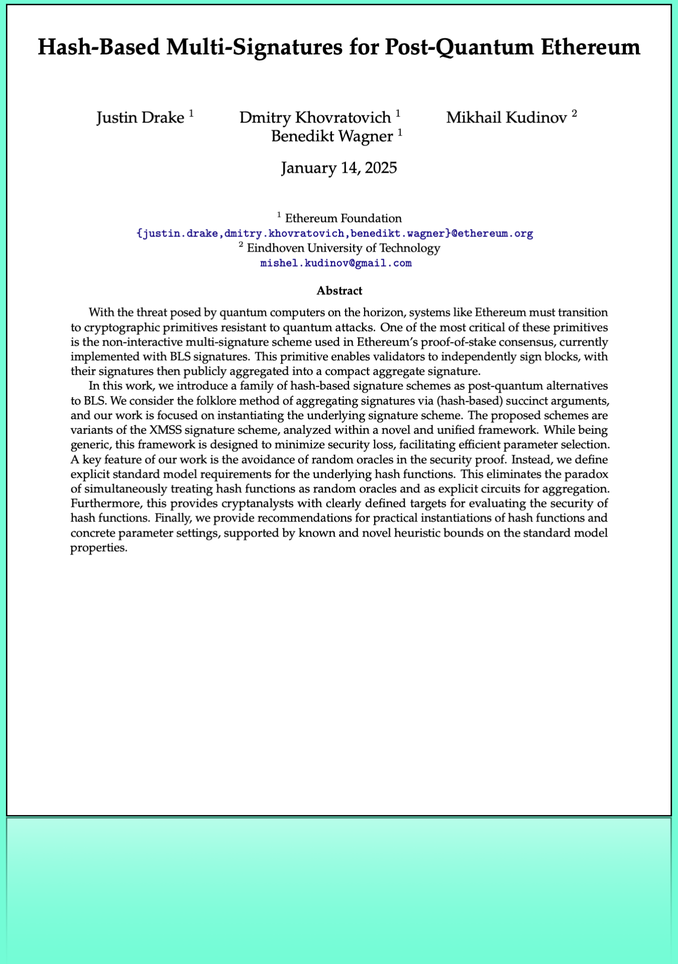
What have we done

What have we done

Paper



Hash-Based Multi-Signatures for Post-Quantum Ethereum

Justin Drake [1] Dmitry Khovratovich [1] Mikhail Kudinov [2]
Benedikt Wagner [1]

January 14, 2025

[1] Ethereum Foundation
{justin.drake,dmitry.khovratovich,benedikt.wagner}@ethereum.org
[2] Eindhoven University of Technology
mishel.kudinov@gmail.com

Abstract

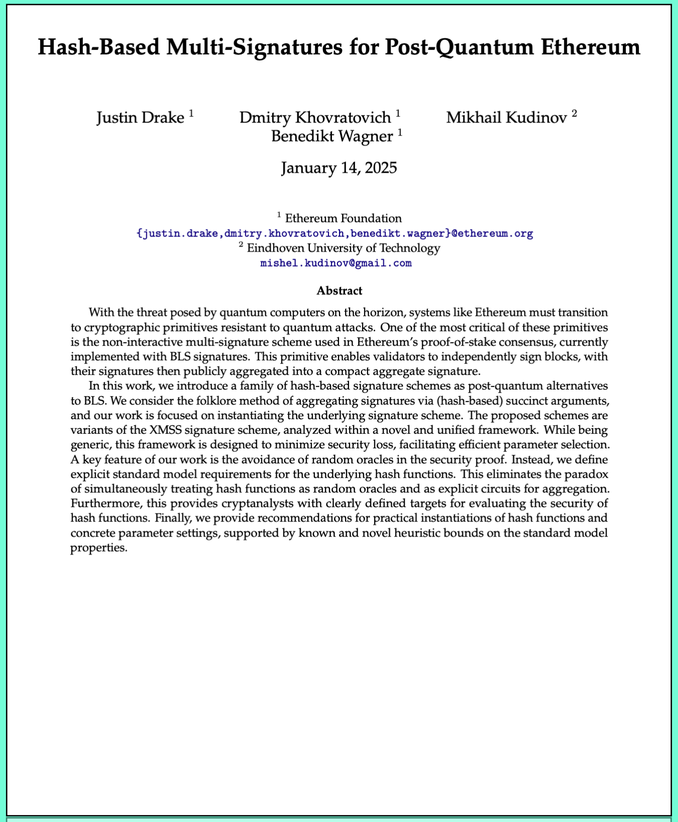
With the threat posed by quantum computers on the horizon, systems like Ethereum must transition to cryptographic primitives resistant to quantum attacks. One of the most critical of these primitives is the non-interactive multi-signature scheme used in Ethereum's proof-of-stake consensus, currently implemented with BLS signatures. This primitive enables validators to independently sign blocks, with their signatures then publicly aggregated into a compact aggregate signature.

In this work, we introduce a family of hash-based signature schemes as post-quantum alternatives to BLS. We consider the folklore method of aggregating signatures via (hash-based) succinct arguments, and our work is focused on instantiating the underlying signature scheme. The proposed schemes are variants of the XMSS signature scheme, analyzed within a novel and unified framework. While being generic, this framework is designed to minimize security loss, facilitating efficient parameter selection. A key feature of our work is the avoidance of random oracles in the security proof. Instead, we define explicit standard model requirements for the underlying hash functions. This eliminates the paradox of simultaneously treating hash functions as random oracles and as explicit circuits for aggregation. Furthermore, this provides cryptanalysts with clearly defined targets for evaluating the security of hash functions. Finally, we provide recommendations for practical instantiations of hash functions and concrete parameter settings, supported by known and novel heuristic bounds on the standard model properties.

Code

What have we done

Paper



Hash-Based Multi-Signatures for Post-Quantum Ethereum

Justin Drake [1] Dmitry Khovratovich [1] Mikhail Kudinov [2]
Benedikt Wagner [1]

January 14, 2025

[1] Ethereum Foundation
{justin.drake,dmitry.khovratovich,benedikt.wagner}@ethereum.org
[2] Eindhoven University of Technology
mishel.kudinov@gmail.com

Abstract

With the threat posed by quantum computers on the horizon, systems like Ethereum must transition to cryptographic primitives resistant to quantum attacks. One of the most critical of these primitives is the non-interactive multi-signature scheme used in Ethereum's proof-of-stake consensus, currently implemented with BLS signatures. This primitive enables validators to independently sign blocks, with their signatures then publicly aggregated into a compact aggregate signature.

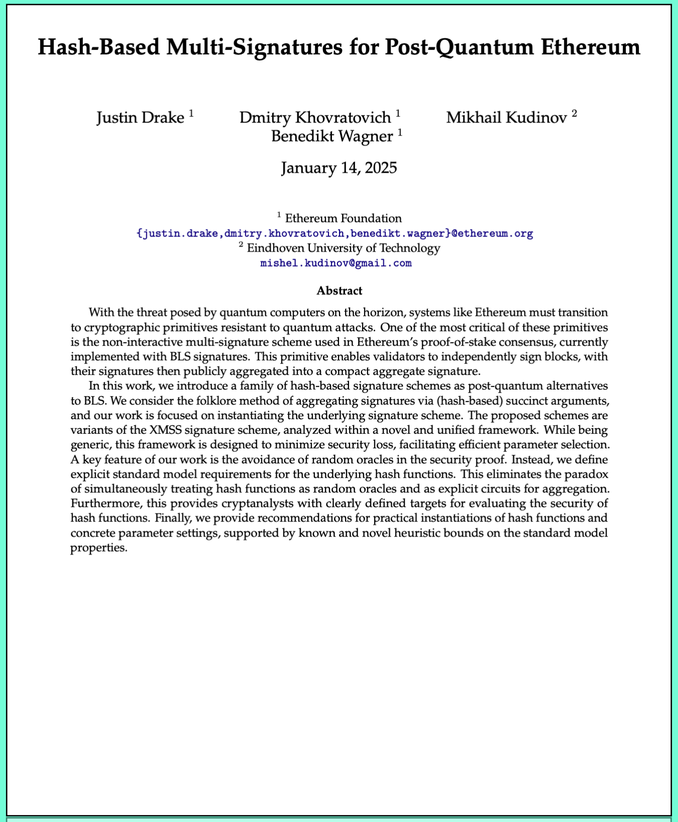
In this work, we introduce a family of hash-based signature schemes as post-quantum alternatives to BLS. We consider the folklore method of aggregating signatures via (hash-based) succinct arguments, and our work is focused on instantiating the underlying signature scheme. The proposed schemes are variants of the XMSS signature scheme, analyzed within a novel and unified framework. While being generic, this framework is designed to minimize security loss, facilitating efficient parameter selection. A key feature of our work is the avoidance of random oracles in the security proof. Instead, we define explicit standard model requirements for the underlying hash functions. This eliminates the paradox of simultaneously treating hash functions as random oracles and as explicit circuits for aggregation. Furthermore, this provides cryptanalysts with clearly defined targets for evaluating the security of hash functions. Finally, we provide recommendations for practical instantiations of hash functions and concrete parameter settings, supported by known and novel heuristic bounds on the standard model properties.

Code

Rigorous Security Analysis

What have we done

Paper



Hash-Based Multi-Signatures for Post-Quantum Ethereum

Justin Drake [1] Dmitry Khovratovich [1] Mikhail Kudinov [2]
Benedikt Wagner [1]

January 14, 2025

[1] Ethereum Foundation
{justin.drake,dmitry.khovratovich,benedikt.wagner}@ethereum.org
[2] Eindhoven University of Technology
mishel.kudinov@gmail.com

Abstract

With the threat posed by quantum computers on the horizon, systems like Ethereum must transition to cryptographic primitives resistant to quantum attacks. One of the most critical of these primitives is the non-interactive multi-signature scheme used in Ethereum's proof-of-stake consensus, currently implemented with BLS signatures. This primitive enables validators to independently sign blocks, with their signatures then publicly aggregated into a compact aggregate signature.

In this work, we introduce a family of hash-based signature schemes as post-quantum alternatives to BLS. We consider the folklore method of aggregating signatures via (hash-based) succinct arguments, and our work is focused on instantiating the underlying signature scheme. The proposed schemes are variants of the XMSS signature scheme, analyzed within a novel and unified framework. While being generic, this framework is designed to minimize security loss, facilitating efficient parameter selection. A key feature of our work is the avoidance of random oracles in the security proof. Instead, we define explicit standard model requirements for the underlying hash functions. This eliminates the paradox of simultaneously treating hash functions as random oracles and as explicit circuits for aggregation. Furthermore, this provides cryptanalysts with clearly defined targets for evaluating the security of hash functions. Finally, we provide recommendations for practical instantiations of hash functions and concrete parameter settings, supported by known and novel heuristic bounds on the standard model properties.

Code

Rigorous Security Analysis

Preliminary Benchmarks

Efficiency

	Encoding	Parameters	Gen [s]	Sign [μs]	Ver [μs]	Sig [KiB]	π_{16} AC	π_{24} AC	π_{16} WC	π_{24} WC
Lifetime $L = 2^{18}$	W	$w = 1$	179.01	362.59	416.54	4.97	81	97	158	97
	W	$w = 2$	168.19	350.04	408.67	2.75	122	59	237	59
	W	$w = 4$	330.52	638.08	769.41	1.66	325	41	615	41
	W	$w = 8$	2717.28	4820	5820	1.11	2917	31	5355	31
	TSW	$w = 1, \delta = 1$	172.67	541.45	396.56	4.75	77	93	77	93
	TSW	$w = 1, \delta = 1.1$	172.29	898.22	376.62	4.75	69	93	69	93
	TSW	$w = 2, \delta = 1$	166.51	530.83	372.93	2.65	117	57	117	57
	TSW	$w = 2, \delta = 1.1$	166.22	888.55	351.37	2.65	105	57	105	57
	TSW	$w = 4, \delta = 1$	312.49	1090.00	650.82	1.58	292	39	292	39
	TSW	$w = 4, \delta = 1.1$	312.64	1670.00	602.75	1.58	263	39	263	39
	TSW	$w = 8, \delta = 1$	2501.01	9760.00	4900.00	1.06	2550	30	2550	30
	TSW	$w = 8, \delta = 1.1$	2499.97	14570.00	4320.00	1.06	2295	30	2295	30
Lifetime $L = 2^{20}$	W	$w = 1$	780.89	362.44	418.31	5.03	82	99	158	99
	W	$w = 2$	705.42	336.30	400.60	2.81	122	61	237	61
	W	$w = 4$	1353.18	617.48	746.28	1.72	326	43	615	43
	W	$w = 8$	11122.95	4981.20	6039.40	1.34	2917	35	5355	35
	TSW	$w = 1, \delta = 1$	752.57	520.42	401.32	4.81	77	95	77	95
	TSW	$w = 1, \delta = 1.1$	731.79	844.01	381.23	4.81	69	95	69	95
	TSW	$w = 2, \delta = 1$	667.76	527.17	379.56	2.7	117	59	117	59
	TSW	$w = 2, \delta = 1.1$	668.14	853.66	354.09	2.7	105	59	105	59
	TSW	$w = 4, \delta = 1$	1249.52	1057.40	661.61	1.64	292	41	292	41
	TSW	$w = 4, \delta = 1.1$	1248.35	1600.00	603.65	1.64	263	41	263	41
	TSW	$w = 8, \delta = 1$	9972.32	9509.50	4870.60	1.27	2550	34	2550	34
	TSW	$w = 8, \delta = 1.1$	9927.97	14271.00	4358.60	1.27	2295	34	2295	34

* Assuming Poseidon2 is used, 128-bit classical security

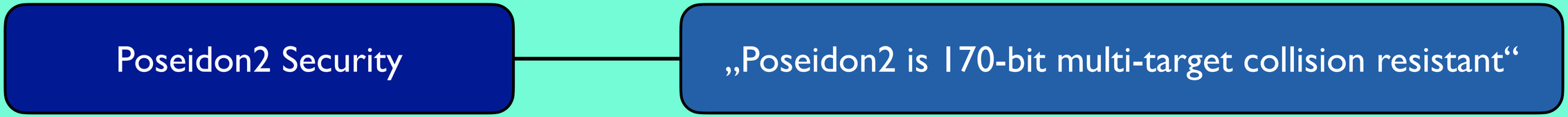
Efficiency

	Encoding	Parameters	Gen [s]	Sign [μs]	Ver [μs]	Sig [KiB]	π_{16} AC	π_{24} AC	π_{16} WC	π_{24} WC
Lifetime $L = 2^{18}$	W	$w = 1$	179.01	362.59	416.54	4.97	81	97	158	97
	W	$w = 2$	168.19	350.04	408.67	2.75	122	59	237	59
	W	$w = 4$	330.52	638.08	769.41	1.66	325	41	615	41
	W	$w = 8$	2717.28	4820	5820	1.11	2917	31	5355	31
	TSW	$w = 1, \delta = 1$	172.67	541.45	396.56	4.75	77	93	77	93
	TSW	$w = 1, \delta = 1.1$	172.29	898.22	376.62	4.75	69	93	69	93
	TSW	$w = 2, \delta = 1$	166.51	530.83	372.93	2.65	117	57	117	57
	TSW	$w = 2, \delta = 1.1$	166.22	888.55	351.37	2.65	105	57	105	57
	TSW	$w = 4, \delta = 1$	312.49	1090.00	650.82	1.58	292	39	292	39
	TSW	$w = 4, \delta = 1.1$	312.64	1670.00	602.75	1.58	263	39	263	39
	TSW	$w = 8, \delta = 1$	2501.01	9760.00	4900.00	1.06	2550	30	2550	30
	TSW	$w = 8, \delta = 1.1$	2499.97	14570.00	4320.00	1.06	2295	30	2295	30
Lifetime $L = 2^{20}$	W	$w = 1$	780.89	362.44	418.31	5.03	82	99	158	99
	W	$w = 2$	705.42	336.30	400.60	2.81	122	61	237	61
	W	$w = 4$	1353.18	617.48	746.28	1.72	326	43	615	43
	W	$w = 8$	11122.95	4981.20	6039.40	1.34	2917	35	5355	35
	TSW	$w = 1, \delta = 1$	752.57	520.42	401.32	4.81	77	95	77	95
	TSW	$w = 1, \delta = 1.1$	731.79	844.01	381.23	4.81	69	95	69	95
	TSW	$w = 2, \delta = 1$	667.76	527.17	379.56	2.7	117	59	117	59
	TSW	$w = 2, \delta = 1.1$	668.14	853.66	354.09	2.7	105	59	105	59
	TSW	$w = 4, \delta = 1$	1249.52	1057.40	661.61	1.64	292	41	292	41
	TSW	$w = 4, \delta = 1.1$	1248.35	1600.00	603.65	1.64	263	41	263	41
	TSW	$w = 8, \delta = 1$	9972.32	9509.50	4870.60	1.27	2550	34	2550	34
	TSW	$w = 8, \delta = 1.1$	9927.97	14271.00	4358.60	1.27	2295	34	2295	34

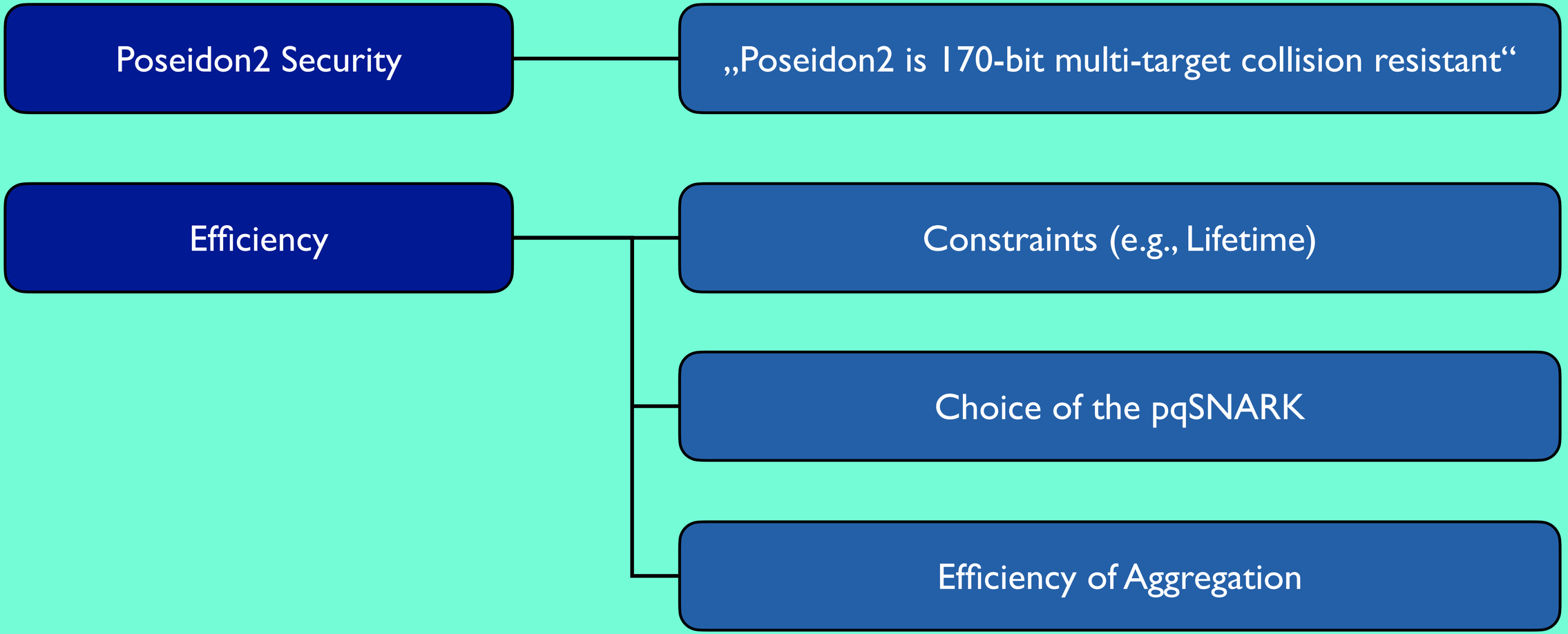
* Assuming Poseidon2 is used, 128-bit classical security

Next Steps

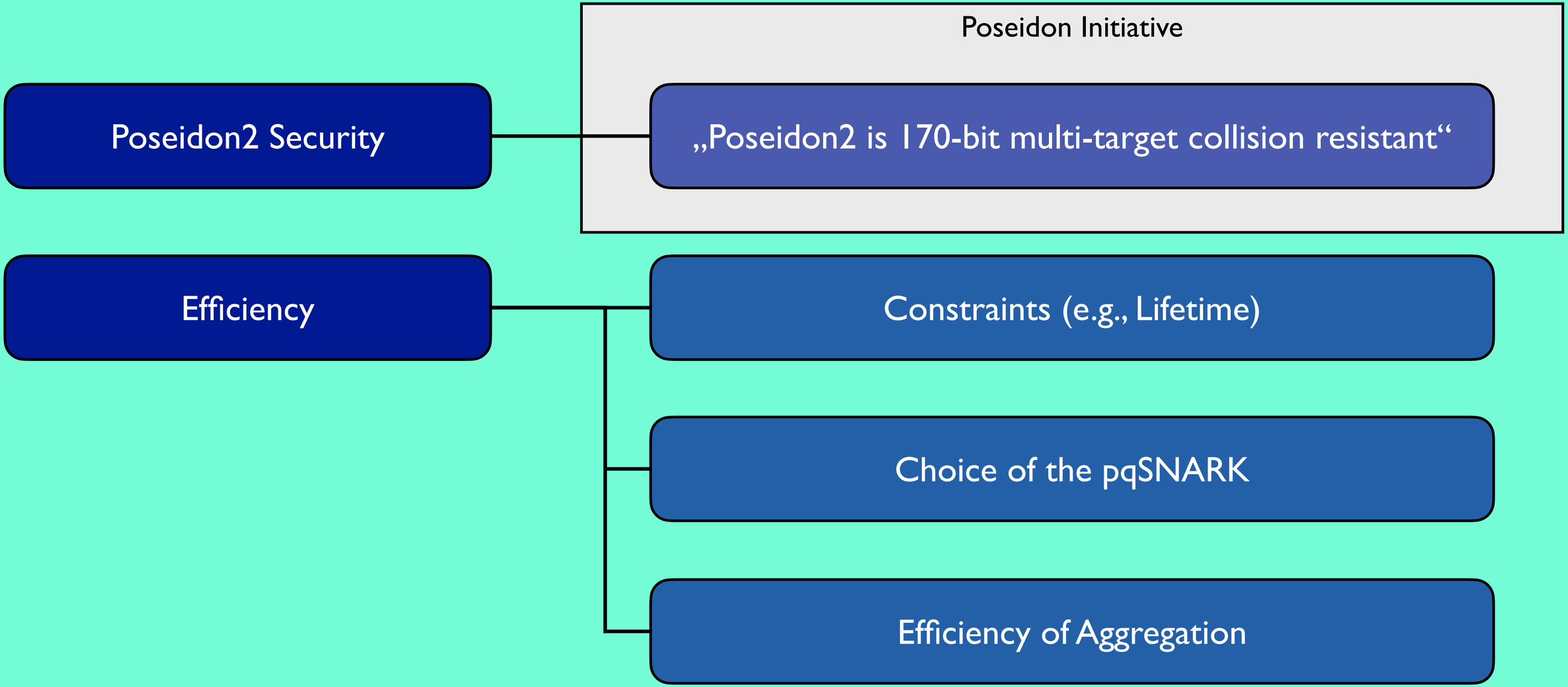
Next Steps


Poseidon2 Security
„Poseidon2 is 170-bit multi-target collision resistant“

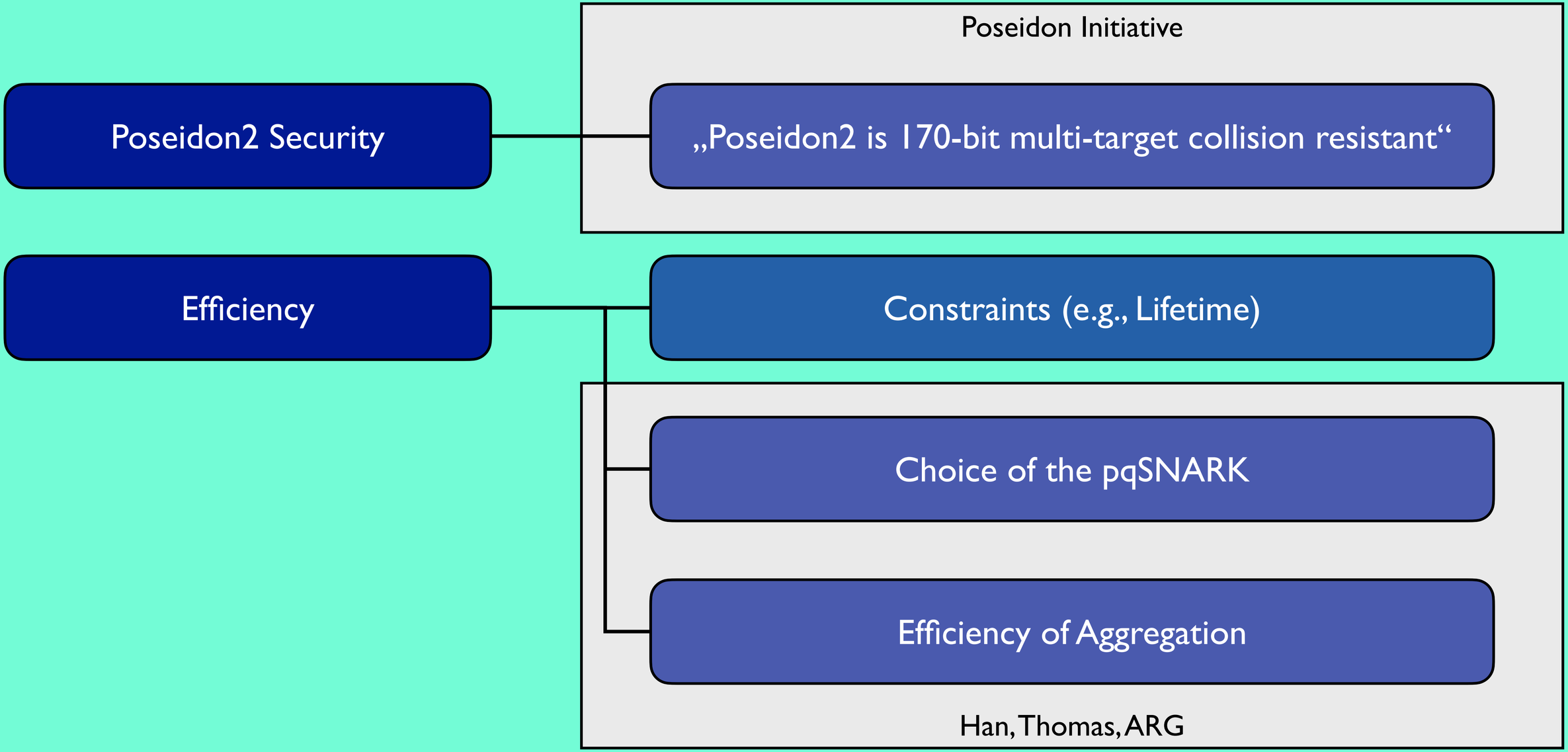
Next Steps


Poseidon2 Security
„Poseidon2 is 170-bit multi-target collision resistant“
Efficiency
Constraints (e.g., Lifetime)
Choice of the pqSNARK
Efficiency of Aggregation

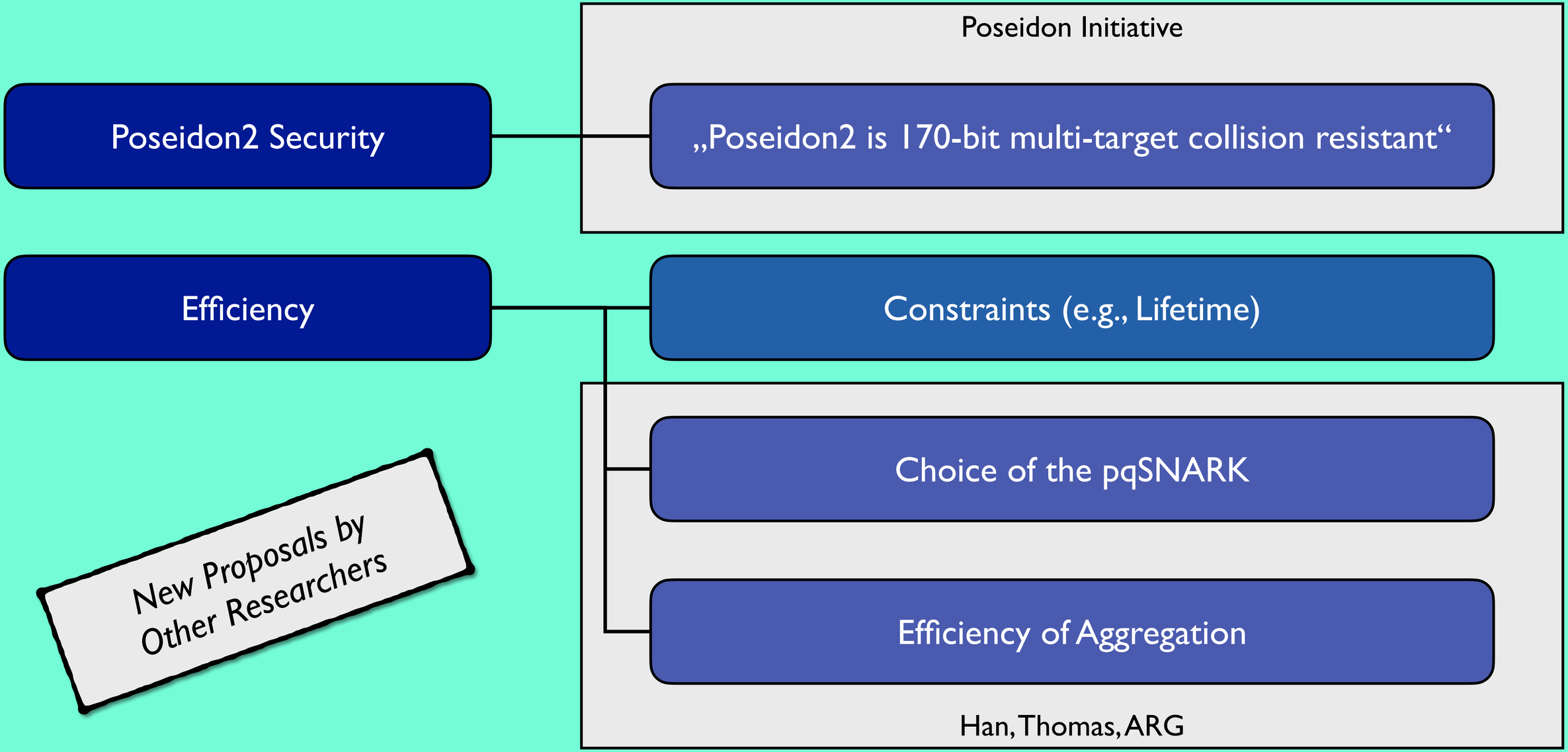
Next Steps


Poseidon Initiative
Poseidon2 Security
„Poseidon2 is 170-bit multi-target collision resistant“
Efficiency
Constraints (e.g., Lifetime)
Choice of the pqSNARK
Efficiency of Aggregation

Next Steps


Poseidon Initiative
Poseidon2 Security
„Poseidon2 is 170-bit multi-target collision resistant“
Efficiency
Constraints (e.g., Lifetime)
Choice of the pqSNARK
Efficiency of Aggregation
Han, Thomas, ARG

Next Steps


Poseidon Initiative
Poseidon2 Security
„Poseidon2 is 170-bit multi-target collision resistant“
Efficiency
Constraints (e.g., Lifetime)
Choice of the pqSNARK
Efficiency of Aggregation
Han, Thomas, ARG
New Proposals by Other Researchers