RUHR-UNIVERSITÄT BOCHUM

A Holistic Approach Towards Side-Channel Secure Fixed-Weight Polynomial Sampling

Markus Krausz, Georg Land, Jan Richter-Brockmann, Tim Güneysu

May 9, 2023

Chair for Security Engineering
Faculty of Computer Science
Ruhr University Bochum

Fixed-Weight Polynomial Sampling (FWPS)

Problem: Generate a random bitstring of length **N** and Hammingweight **W**.

Example: **N** = 10, **W** = 4

1	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**0**

Representations:

Coefficient Representation:

1	0	0	1	0	0	1	1	0	0

Index Representation:

0	3	6	7

Fixed-Weight Polynomial Sampling (FWPS)

Problem: Draw **W** random elements from the set {0, …, **N**-1} without replacement, regardless of the order.

Example: **N** = 10, **W** = 4

0	**3**	**6**	**7**

Why do we need this?

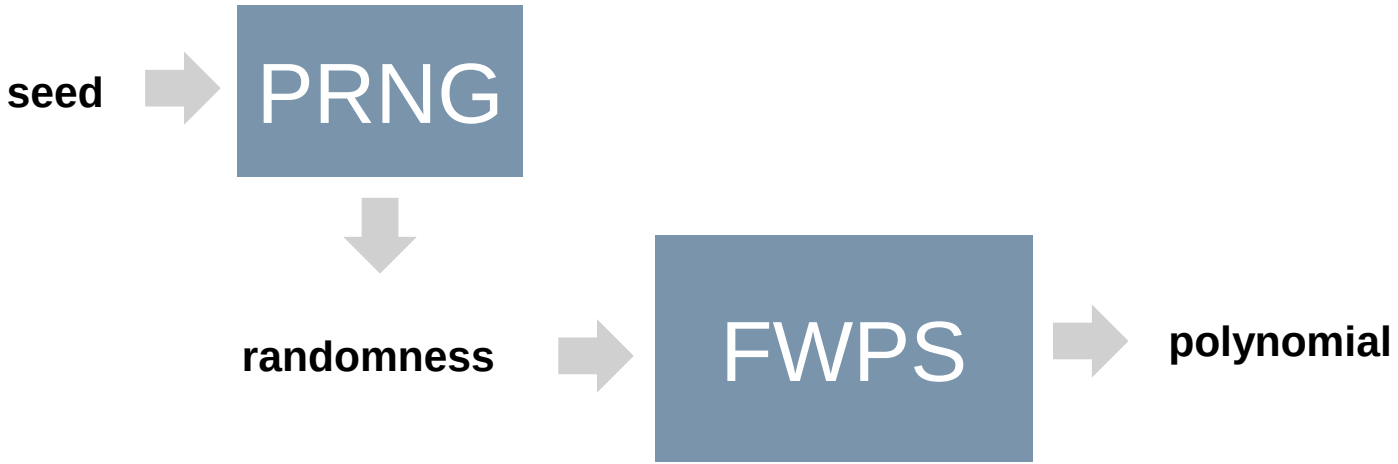
Post Quantum Cryptography:

Scheme	N	W	W/N
BIKE	49318	199	0.4 %
HQC	35851	114	0.4 %
McEliece	6688	128	2.0 %
NTRU	677	254	37.5 %
sNTRU Prime	953	396	41.6 %

Side-Channel Security

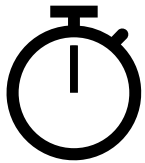
Side-Channel Security

Secret Values



Side-Channel Security

Countermeasues



Timing Side-Channels

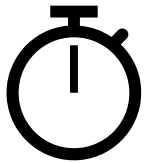
Constant-Time Programming

Power Side-Channels

Masking

Side-Channel Security

Constant-Time



Timing

Side-Channels

Constant-Time Programming:

- branches
- memory accesses
- instructions with operand dependent runtime

must NOT depend on secret values

Side-Channel Countermeasures

Masking

Power

Side-Channels

Masking:

secret values must be split into random shares

Boolean: $s = s0 \oplus s1$

Methods for FWPS

Method	Category	Representation	Seed Secure
Simple Index Rejection	Rejection Sampling	Index	No
Bounded Index Rejection	Rejection Sampling	Index	Yes
ANDing	Setting Bits randomly	Coefficient	No
Comparison	Setting Bits randomly	Coefficient	No
CT Fisher-Yates	Shuffling	Index	Yes
Sorting	Shuffling	Coefficient	Yes

Rejection Sampling

Category 1: Index Rejection Sampling

Simple Method

Idea:
Sample uniform random values from [0, N),
reject a value if we have a collision,
repeat until we have W distinct values .

Category 1: Index Rejection Sampling

Bounded Method

Idea:
Determine an upper bound B of iterations in which with very high probability, enough distinct values will be sampled.

Always iterate over the whole array to keep the count secret.

Setting Random Bits

Category 2: Setting Random Bits

Repeated ANDing

Idea: Start with a random polynomial of length N and adapt the weight by ANDing and ORing random polynomials until W is reached.

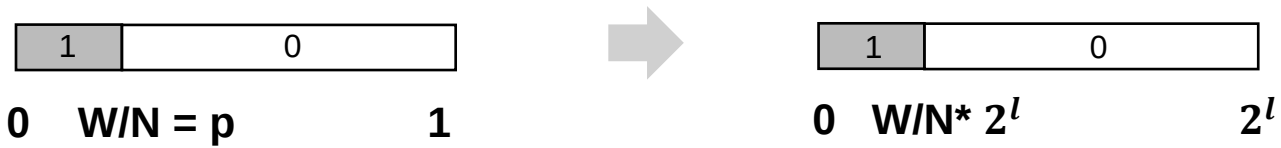
1	0	1	1	0	0	1	0	0	1	W = 5
				AND						
0	1	1	0	0	1	1	1	0	1	W = 6
				=						
0	0	1	0	0	0	1	0	0	1	W = 3

Category 2: Setting Random Bits

Comparison

Idea: For each coefficent set a bit with probability p = W/N,
resample polynomial until one with weight W is generated.

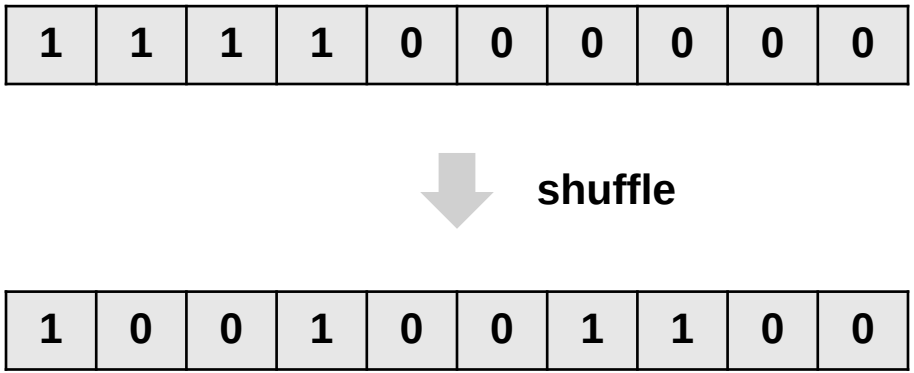
Approximate p with a comparison of a random l-bit string with $\lceil W/N * 2^l \rceil$



Shuffling

Category 3: Shuffling

Idea: Start with fix polynomial of length N with correct weight W and apply a random permutation.



Category 3: Shuffling

Sorting

Idea: Pair each coefficient with a distinct random number, sort the pairs according to the random numbers to get a random permutation of the coefficients

1	1	1	1	0	0	0	0	0	0
14	0	61	50	67	8	13	34	36	74

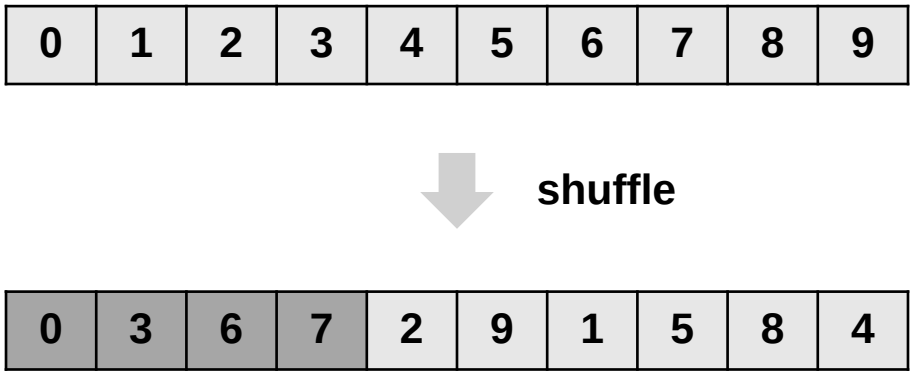
sort

1	0	0	1	0	0	1	1	0	0
0	8	13	14	34	36	50	61	67	74

Category 3: Shuffling

Constant Time Fisher-Yates

Idea: Start with an array of the values 0, …, N-1,
apply a random permutation and
take the first W entries as a polynomial in index representation.



Evaluation

Cortex-M4, first order masked, kilo cycles

Scheme	N	W	N/W	Sort	F-Y	Rej	B Rej	AND	Comp
BIKE	49318	199	0.4 %	OOM	13206	-	69140	-	-
HQC	35851	114	0.4 %	OOM	5377	-	41500	-	-
McEliece	6688	128	2.0 %	240949	5044	1555	-	63766	31652
NTRU	677	254	37.5 %	14958	12445	4833	-	3559	2935
sNTRU Pr	953	396	41.6 %	21564	27494	11253	-	4403	6266

Thank you!

Markus Krausz

Chair for Security Engineering
Faculty of Computer Science
Ruhr University Bochum

Masked Core Operations

Conditional Move

If (c) : d = s

naive: $d = (d \wedge \neg c) \vee (s \wedge c)$

faster: $d = d \oplus ((d \oplus s) \wedge c)$

Masked Core Operations

Comparison a < b

Classic Solution: Subtract (Ripple-Carry) and check if result negativ:

$$
\begin{aligned}
r[0] &= a[0] \oplus b[0] \\
c[0] &= \overline{a[0]} \wedge b[0] \\
r[i] &= a[i] \oplus b[i] \oplus c[i-1] \quad \forall 1 \leq i < l \\
c[i] &= (c[i-1] \wedge (\overline{a[i]} \oplus b[i])) \oplus (\overline{a[i]} \wedge b[i]) \quad \forall 1 \leq i < l \\
r[l] &= c[l-1]
\end{aligned}
$$

Masked Core Operations

Comparison a < b

Faster Approach:

$$a = 1\ 1\ 1\ 0$$
$$b = 1\ 0\ 1\ 1$$
$$a \oplus b = c = 0\ 1\ 0\ 1$$

Parameters

Scheme	Param.	Where?	N	W	W/N	Target Space	Det.	Sec. Seed
BIKE	L1	en/decaps	24646	134	0.005	binary	yes	yes
BIKE	L1	keygen	12323	71	0.006	binary	no	no
BIKE	L3	en/decaps	49318	199	0.004	binary	yes	yes
BIKE	L3	keygen	24659	103	0.004	binary	no	no
BIKE	L5	en/decaps	81194	264	0.003	binary	yes	yes
BIKE	L5	keygen	40973	137	0.003	binary	no	no
HQC	128	en/decaps	17669	75	0.004	binary	yes	yes
HQC	128	keygen	17669	66	0.004	binary	no	no
HQC	192	en/decaps	35851	114	0.003	binary	yes	yes
HQC	192	keygen	35851	100	0.003	binary	no	no
HQC	256	en/decaps	57637	149	0.003	binary	yes	yes
HQC	256	keygen	57637	131	0.003	binary	no	no
McEliece	348864	encaps	3488	64	0.018	binary	no	no
McEliece	460896	encaps	4608	96	0.021	binary	no	no
McEliece	6688128	encaps	6688	128	0.019	binary	no	no
McEliece	6960119	encaps	6960	119	0.017	binary	no	no
McEliece	8192128	encaps	8192	128	0.016	binary	no	no
NTRU	hps2048509	keygen	509	254	0.499	$W/2$ ternary	no	no
NTRU	hps2048677	keygen	677	254	0.375	$W/2$ ternary	no	no
NTRU	hps4096821	keygen	821	510	0.379	$W/2$ ternary	no	no
sNTRU Prime	653	keygen	653	288	0.441	uni. ternary	no	no
NTRU LPRrime	653	keygen	653	252	0.386	uni. ternary	no	no
sNTRU Prime	761	keygen	761	286	0.376	uni. ternary	no	no
NTRU LPRrime	761	keygen	761	250	0.329	uni. ternary	no	no
sNTRU Prime	857	keygen	857	322	0.376	uni. ternary	no	no
NTRU LPRrime	857	keygen	857	329	0.384	uni. ternary	no	no
sNTRU Prime	953	keygen	953	396	0.416	uni. ternary	no	no
NTRU LPRrime	953	keygen	953	345	0.362	uni. ternary	no	no
sNTRU Prime	1013	keygen	1013	448	0.442	uni. ternary	no	no
NTRU LPRrime	1013	keygen	1013	392	0.387	uni. ternary	no	no
sNTRU Prime	1277	keygen	1277	492	0.385	uni. ternary	no	no
NTRU LPRrime	1277	keygen	1277	429	0.336	uni. ternary	no	no

Evaluation

Cortex-M4

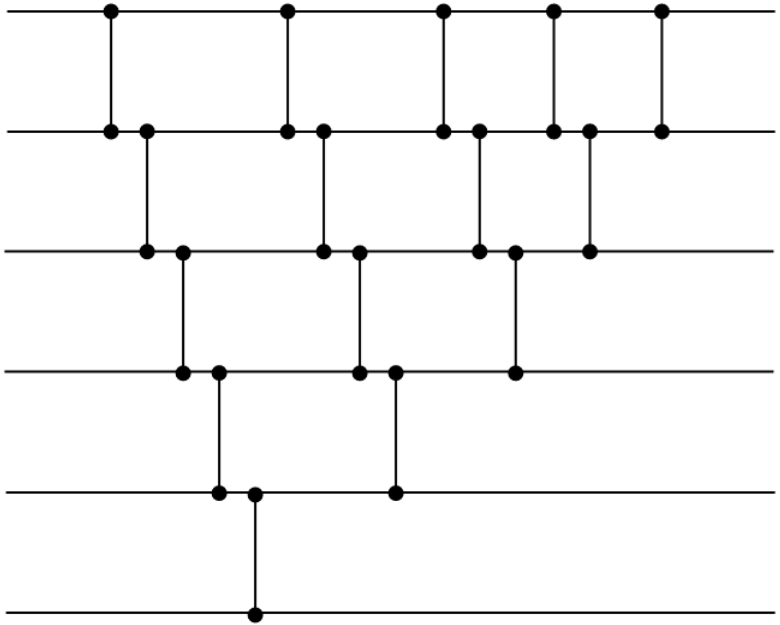
first order masking

kilo cycles

Scheme	N	W	Sort	Fisher-Y.	Reject	B. Reject	RepAND	Comp.	I2C Trans.
BIKE	24646	134	–	7128	–	34077	–	–	770708
BIKE	12323	71	–	2854	647	–	101629*	45838	195945
BIKE	49318	199	–	13206	–	69140	–	–	2394245
BIKE	24659	103	–	4901	1255	–	156631*	129050	592411
BIKE	81194	264	–	21680	–	131135	–	–	5497931
BIKE	40973	137	–	7514	2176	–	320522*	234007	1372560
HQC	17669	75	–	3063	–	25803	–	–	309894
HQC	17669	66	–	2852	620	–	185348*	63242	272707
HQC	35851	114	–	5377	–	41500	–	–	999526
HQC	35851	100	–	5034	1282	–	391503*	183833	876778
HQC	57637	149	–	7808	–	28930	–	–	2094589
HQC	57637	131	–	7367	2132	–	837777*	348099	1841552
McEliece	3488	64	108596	1847	462	–	19519†	12948	32246
McEliece	4608	96	160777	3326	972	–	31778†	20392	68236
McEliece	6688	128	240949	5044	1555	–	63766†	31652	131539
McEliece	6960	119	249618	4848	1386	–	59875†	34571	127568
McEliece	8192	128	300713	4591	1527	–	62867†	34609	161312
NTRU	509	254	9699	11532	4709	–	2141	1666	15342
NTRU	677	254	14958	12445	4833	–	3559	2935	22674
NTRU	821	510	18338	17737	7022	–	4140	3921	32655
sNTRU Prime	653	288	14958	15086	6345	–	3023	3033	24650
NTRU LPRrime	653	252	14958	12390	4806	–	3177	3299	21515
sNTRU Prime	761	286	16464	15063	6005	–	3699	3336	27828
NTRU LPRrime	761	250	16464	12350	4570	–	3457	3773	24264
sNTRU Prime	857	322	19848	20249	7461	–	4125	4012	34948
NTRU LPRrime	857	329	19848	20569	7805	–	4482	4650	35825
sNTRU Prime	953	396	21564	27494	11253	–	4403	6266	47664
NTRU LPRrime	953	345	21564	20867	8404	–	4496	6617	41385
sNTRU Prime	1013	448	23405	31680	14421	–	4836	5763	57395
NTRU LPRrime	1013	392	23405	27380	10843	–	5428	7015	50228
sNTRU Prime	1277	492	32361	42388	18445	–	6861	9612	85013
NTRU LPRrime	1277	429	32361	33673	13822	–	7285	9245	74318

Approach 3: Shuffling

Sorting Networks



Approach 3: Shuffling

Bitonicsort