



Enhancing Code Based Zero-knowledge Proofs

using Rank Metric

16/12/2020

Emanuele Bellini¹ Philippe Gaborit² Alexandros Hasikos^{1,3} Victor Mateu¹

¹TII Cryptography Research Centre ²University of Limogés ³Universitat Pompeu Fabra

Contents



1. Contributions

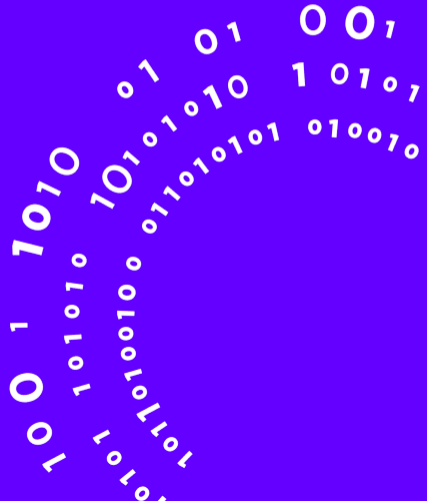
2. Preliminaries

3. Commitment Scheme

4. Performance

5. Conclusions and Future Work

Contributions



Contributions of this work



- Adapt Jain et al. (2012) work and design a perfectly binding and computationally hiding commitment scheme based on the Rank Syndrome Decoding (RSD) Problem.

Contributions of this work



- Adapt Jain et al. (2012) work and design a perfectly binding and computationally hiding commitment scheme based on the Rank Syndrome Decoding (RSD) Problem.
- Design interactive protocols for:
 - Knowledge of valid opening.
 - Proving linear relations.
 - Proving multiplicative (or any bitwise) relations.

Contributions of this work



- Adapt Jain et al. (2012) work and design a perfectly binding and computationally hiding commitment scheme based on the Rank Syndrome Decoding (RSD) Problem.
- Design interactive protocols for:
 - Knowledge of valid opening.
 - Proving linear relations.
 - Proving multiplicative (or any bitwise) relations.
- Compute secure parameters for both LPN and RSD variants of the protocols.

Contributions of this work



- Adapt Jain et al. (2012) work and design a perfectly binding and computationally hiding commitment scheme based on the Rank Syndrome Decoding (RSD) Problem.
- Design interactive protocols for:
 - Knowledge of valid opening.
 - Proving linear relations.
 - Proving multiplicative (or any bitwise) relations.
- Compute secure parameters for both LPN and RSD variants of the protocols.
- Implement and compare (performance and efficiency) of both LPN and RSD variants with suggested parameters for 128 bits of security.

Highlights



- Our work is the first zero-knowledge protocol for arbitrary circuits whose security relies on the Rank Syndrome Decoding problem.

Highlights



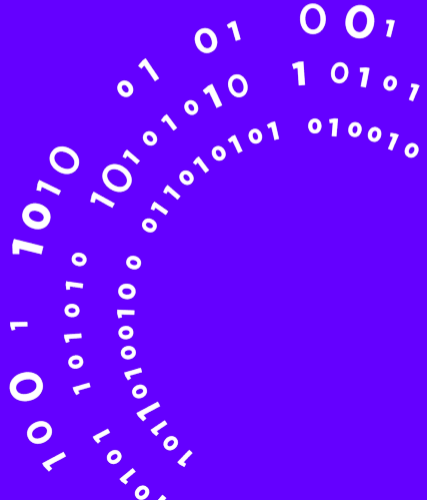
- Our work is the first zero-knowledge protocol for arbitrary circuits whose security relies on the Rank Syndrome Decoding problem.
- Our proposal (RSD) generates proofs that are 60% smaller than the LPN variant for the same security level.

Highlights



- Our work is the first zero-knowledge protocol for arbitrary circuits whose security relies on the Rank Syndrome Decoding problem.
- Our proposal (RSD) generates proofs that are 60% smaller than the LPN variant for the same security level.
- Public parameters of the RSD variant are only 1% of respective parameters for the LPN variant.

Preliminaries



Definitions

Definition (Linear $(n, k)_q$ -code)

A linear $(n, k)_q$ -code C is a vector subspace of $(\mathbb{F}_q)^n$ of dimension k , where k and n are positive integers such that $k < n$, q is a prime power, and \mathbb{F}_q is the finite field with q elements. Elements of the vector space are called **vectors** or **words**, while elements of the code are called **codewords**.

Definition (Linear $(n, k)_q$ -code)

A linear $(n, k)_q$ -code C is a vector subspace of $(\mathbb{F}_q)^n$ of dimension k , where k and n are positive integers such that $k < n$, q is a prime power, and \mathbb{F}_q is the finite field with q elements. Elements of the vector space are called **vectors** or **words**, while elements of the code are called **codewords**.

Definition (Generator and Parity Check Matrices)

A matrix $G \in \mathcal{M}_{k,n}^*(\mathbb{F}_q)$ is called a generator matrix of C if its rows form a basis of C , i.e.

$C = \{x \cdot G : x \in (\mathbb{F}_q)^k\}$. A matrix $H \in \mathcal{M}_{n-k,n}^*(\mathbb{F}_q)$ is called a parity-check matrix of C if

$C = \{x \in (\mathbb{F}_q)^n : H \cdot x^T = 0\}$

Definitions



Definition (Hamming weight $w_H(v)$ of a vector)

The hamming weight $w_H(v)$ of a vector v is the number of its non-zero bits.

Definitions



Definition (Hamming weight $w_H(v)$ of a vector)

The hamming weight $w_H(v)$ of a vector v is the number of its non-zero bits.

Definition (Rank weight $w_R(v)$ of a vector)

The rank weight $w_R(v)$ of a vector v is the rank of its matrix representation (number of linearly independent vectors).

Definitions

Definition (Rank preserving transformation function $\Pi_{P,Q}(v)$)

Let $Q \in \mathcal{M}_{m,m}^*(\mathbb{F}_q)$ be a q -ary matrix of size $m \times m$, $P \in \mathcal{M}_{n,n}^*(\mathbb{F}_q)$ be a q -ary matrix of size $n \times n$, and $v = (v_1, \dots, v_n) \in (\mathbb{F}_{q^m})^n$. We define the function $\Pi_{P,Q}$ such that $(\pi_1, \dots, \pi_n) = \Pi_{P,Q}(v) = \phi^{-1}(Q \cdot \phi(v) P) \in (\mathbb{F}_{q^m})^n$, where for $h = 1, \dots, n$, $\pi_h := \beta_1 \sum_{i=1}^m \sum_{j=1}^n Q_{1,i} v_{i,j} P_{j,h} + \dots + \beta_m \sum_{i=1}^m \sum_{j=1}^n Q_{m,i} v_{i,j} P_{j,h}$, with $\beta = \{\beta_1, \dots, \beta_m\}$ a basis of $(\mathbb{F}_q)^m$

Definition (Rank preserving transformation function $\Pi_{P,Q}(v)$)

Let $Q \in \mathcal{M}_{m,m}^*(\mathbb{F}_q)$ be a q -ary matrix of size $m \times m$, $P \in \mathcal{M}_{n,n}^*(\mathbb{F}_q)$ be a q -ary matrix of size $n \times n$, and $v = (v_1, \dots, v_n) \in (\mathbb{F}_{q^m})^n$. We define the function $\Pi_{P,Q}$ such that $(\pi_1, \dots, \pi_n) = \Pi_{P,Q}(v) = \phi^{-1}(Q \cdot \phi(v) P) \in (\mathbb{F}_{q^m})^n$, where for $h = 1, \dots, n$, $\pi_h := \beta_1 \sum_{i=1}^m \sum_{j=1}^n Q_{1,i} v_{i,j} P_{j,h} + \dots + \beta_m \sum_{i=1}^m \sum_{j=1}^n Q_{m,i} v_{i,j} P_{j,h}$, with $\beta = \{\beta_1, \dots, \beta_m\}$ a basis of $(\mathbb{F}_q)^m$

Gaborit et al. (2011) proved that:

- For any $x, y \in (\mathbb{F}_{q^m})^n$ any full rank $P \in \mathcal{M}_{n,n}^*(\mathbb{F}_q)$ and $Q \in \mathcal{M}_{m,m}^*(\mathbb{F}_q)$
- $\Pi_{P,Q}$ has the rank preserving property $w_R(\Pi_{P,Q}(x)) = w_R(x)$ and is a linear mapping $a\Pi_{P,Q}(x) + b\Pi_{P,Q}(y) = \Pi_{P,Q}(ax + by)$.
- $\Pi_{P,Q}$ is invertible if P and Q are.

Decoding Problem



The decoding problem for random linear codes, consists of searching for the closest codeword to a given vector:

Decoding Problem



The decoding problem for random linear codes, consists of searching for the closest codeword to a given vector:

Decoding Problem

Given G , $y = xG + e$, and the weight w , find the pair (x, e) , where the weight of e is w .

Decoding Problem



The decoding problem for random linear codes, consists of searching for the closest codeword to a given vector:

Decoding Problem

Given G , $y = xG + e$, and the weight w , find the pair (x, e) , where the weight of e is w .

In the case of random linear codes, the decoding problem is equivalent to the syndrome decoding problem:

Decoding Problem



The decoding problem for random linear codes, consists of searching for the closest codeword to a given vector:

Decoding Problem

Given G , $y = xG + e$, and the weight w , find the pair (x, e) , where the weight of e is w .

In the case of random linear codes, the decoding problem is equivalent to the syndrome decoding problem:

Syndrome Decoding Problem

Given H , $s = Hy$, and the weight w , find y , where the hamming weight of y is w .

Decoding Problem



The decoding problem for random linear codes, consists of searching for the closest codeword to a given vector:

Decoding Problem

Given G , $y = xG + e$, and the weight w , find the pair (x, e) , where the weight of e is w .

In the case of random linear codes, the decoding problem is equivalent to the syndrome decoding problem:

Syndrome Decoding Problem

Given H , $s = Hy$, and the weight w , find y , where the hamming weight of y is w .

The Rank Syndrome Decoding problem is the same as the Syndrome Decoding problem however the metric used for the weight of the error is the **rank** instead of the Hamming weight.

Definition (Commitment Schemes)

A triple of algorithms (**Setup**, **Com**, **Ver**) is called a commitment scheme if it satisfied the following:

- On input 1^ℓ the setup algorithm **Setup** outputs the public commitment parameters **pp**.
- The commitment algorithm **Com** takes as input a message **m** from a message space **M** and the public commitment parameters **pp**, and outputs a commitment/opening pair (**c**, **d**).
- The verification algorithm **Ver** take the parameters **pp**, a message **m**, a commitment **c** and an opening **d** and outputs **true** or **false**.

Properties of commitment schemes



The commitment scheme we will describe satisfies the following security properties:

- *Correctness*: **Ver** evaluates to **true** if the inputs are honestly computed.

Properties of commitment schemes



The commitment scheme we will describe satisfies the following security properties:

- *Correctness*: **Ver** evaluates to **true** if the inputs are honestly computed.
- *Perfect Binding*: With overwhelming probability over the choice of the public commitment parameters, no commitment can be opened in two different ways.

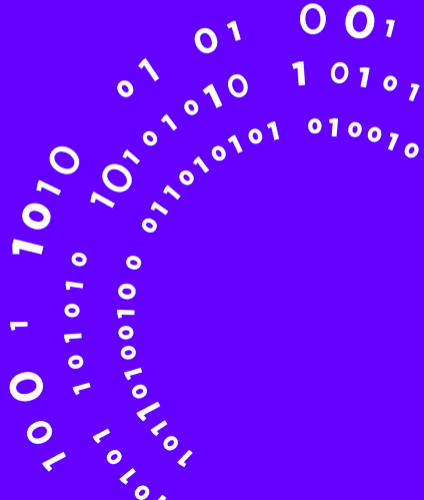
Properties of commitment schemes



The commitment scheme we will describe satisfies the following security properties:

- *Correctness*: **Ver** evaluates to **true** if the inputs are honestly computed.
- *Perfect Binding*: With overwhelming probability over the choice of the public commitment parameters, no commitment can be opened in two different ways.
- *Computational Hiding*: A commitment, computationally hides the committed message if the commitments are computationally indistinguishable.

Commitment Scheme



Commitment scheme in the rank metric



Let q be the prime characteristic, m the degree of the q -ary extension field \mathbb{F}_{q^m} , the bit length μ of a message $m \in \mathbb{F}_q^\mu$, the bit length π of the randomness $s \in \mathbb{F}_q^\pi$, the length n of the linear code C , and the rank weight ρ of an error $e \in \mathbb{F}_{q^m}^n$.

Commitment scheme in the rank metric



Let q be the prime characteristic, m the degree of the q -ary extension field \mathbb{F}_{q^m} , the bit length μ of a message $m \in \mathbb{F}_q^\mu$, the bit length π of the randomness $s \in \mathbb{F}_q^\pi$, the length n of the linear code C , and the rank weight ρ of an error $e \in \mathbb{F}_{q^m}^n$.

Setup(1^ℓ)

$$G_m \leftarrow \mathcal{M}_{\frac{\mu}{m}, n}^* (\mathbb{F}_{q^m})$$

$$G_s \leftarrow \mathcal{M}_{\frac{\pi}{m}, n}^* (\mathbb{F}_{q^m})$$

$$\text{return } G = (G_s^T \| G_m^T)^T$$

Com $_G(m)$

$$s \leftarrow_{\$} \mathbb{F}_2^\pi$$

$$e \leftarrow_{\$} \mathbb{F}_{q^m}^n, \text{ s.t. } w_R(e) = \rho$$

$$c = (s \| m) \cdot G + e$$

return c, s

Ver $_G(c, m', s')$

$$e' = c + (s' \| m') \cdot G$$

if $w_R(e') = \rho$ return True

else return False

Parameters



In order to compare our scheme (RSD) with the Hamming weight variant (LPN) we had to compute parameters for both.

Parameters

In order to compare our scheme (RSD) with the Hamming weight variant (LPN) we had to compute parameters for both.

For a quantum security level of 128 bits:

		Parameters	Secret	Public Param.	Average Communication
Hamming	Formula	(n, k, w)	$k + n$	$n + kn + \log_2(w)$	$5n + \lceil 2/3(n \log_2(n)) \rceil + 2\lambda$
	Bits	(2640,1320,284)	3960	3487449	33461
Rank (this work)	Formula	(q, m, n, k, ρ)	$mk + mn$	$mn + mkn + \log_2(\rho)$	$5mn + \lceil 2/3(m^2 + n^2) \rceil + 2\lambda$
	Bits	(2,43,38,17,13)	2365	29416	10622

Table 1: Communication cost and parameters bit sizes of the Σ -protocol of knowledge of valid opening

Parameters

In order to compare our scheme (RSD) with the Hamming weight variant (LPN) we had to compute parameters for both.

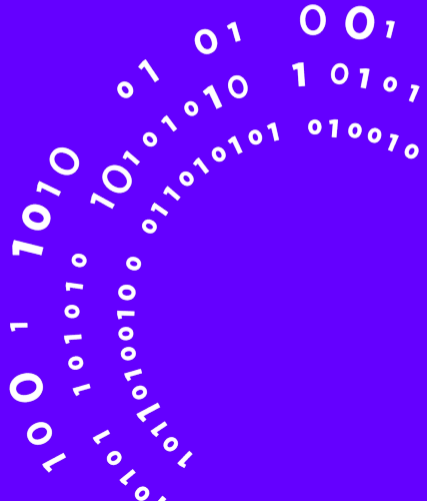
For a quantum security level of 128 bits:

		Parameters	Secret	Public Param.	Average Communication
Hamming	Formula	(n, k, w)	$k + n$	$n + kn + \log_2(w)$	$5n + \lceil 2/3(n \log_2(n)) \rceil + 2\lambda$
	Bits	(2640,1320,284)	3960	3487449	33461
Rank (this work)	Formula	(q, m, n, k, ρ)	$mk + mn$	$mn + mkn + \log_2(\rho)$	$5mn + \lceil 2/3(m^2 + n^2) \rceil + 2\lambda$
	Bits	(2,43,38,17,13)	2365	29416	10622

Table 1: Communication cost and parameters bit sizes of the Σ -protocol of knowledge of valid opening

Average communication cost is about 60% lower while the public parameters size is two orders of magnitude lower. The size of the secret in ZKP is 40% lower.

Performance



Implementation



We have implemented both the work from Jain et al. (2012) and our variant with the parameters shown in previous slide.

- Implemented in C++ using the NTL library from Victor Shoup.
- Benchmarks conducted on 2.9GHz Quad-Core Intel Core i7 with 16GB of LPDD3 RAM at 2133MHz.
- You can access the code https://github.com/Crypto-TII/2020-CANS-rank_commitments

Commitment Scheme

Commitment Scheme					
Jain et. al.			This work		
Routine	Subroutine	Time [ms]	Routine	Subroutine	Time [ms]
Setup	Generate matrix A	1.303	Setup	Generate matrix G	0.030
Commitment	Generate random vector r	negl.	Commitment	Generate random vector π	negl.
	Generate error vector e	0.168		Generate error vector e	1.800
	Compute commitment c	0.029		Compute commitment c	0.025
	Total	0.197		Total	1.825
Verification	Recover error vector e	0.029	Verification	Recover error vector e	0.0250
	Compute hamming weight of e	0.001		Compute rank of e	0.0160
	Total	0.030		Total	0.041

Table 2: Commitment scheme performance comparison.

Knowledge of Valid Opening

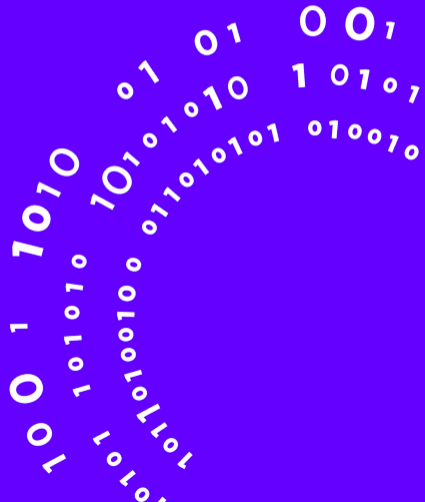
Knowledge of Valid Opening							
Jain et.al.			This work				
Routine	Subroutine	Time [ms]	Routine	Subroutine	Time[ms]		
Proof gen.	Generate π	0.552	Proof gen.	Generate $\Pi_{P,Q}$	0.135		
	Generate random vectors	negl.		Generate random vectors	negl.		
	Comm. 0	t_0		0.032	Comm. 0	r_0	0.020
		$E(t_\pi, t_0)$		0.400		$E(r_{P,Q}, r_0)$	0.035
		$Com(E(t_\pi, t_0))$		0.200		$Com(E(r_{P,Q}, r_0))$	1.860
	Comm. 1	t_1		0.038	Comm. 1	r_1	0.044
		$E(t_1)$		0.391		$E(r_1)$	0.019
		$Com(E(t_1))$		0.203		$Com(E(r_1))$	1.809
	Comm. 2	t_2		0.040	Comm. 2	r_2	0.044
		$E(t_2)$		0.396		$E(r_2)$	0.018
		$Com(E(t_2))$		0.197		$Com(E(r_2))$	1.736
		Total		1.897		Total	5.585
Proof ver.	Verif. 0	$Ver(c_0, E(t_\pi, t_0), s_0)$	0.423	Verif. 0	$Ver(c_0, E(r_{P,Q}, r_0), s_0)$	0.077	
		$Ver(c_1, E(t_1), s_1)$	0.426		$Ver(c_1, E(r_1), s_1)$	0.064	
		$t_0 + \pi^{-1}(t_1) \in \text{Img}(A)$	170.888		$r_0 + \Pi_{r_0}^{-1}(r_1) \in \text{Img}(G)$	2.559	
	Verif. 1	$Ver(c_0, E(t_\pi, t_0), s_0)$	0.424	Verif. 1	$Ver(c_0, E(r_{P,Q}, r_0), s_0)$	0.080	
		$Ver(c_2, E(t_2), s_2)$	0.444		$Ver(c_2, E(r_2), s_2)$	0.066	
		$t_0 + \pi^{-1}(t_2) + y \in \text{Img}(A)$	175.526		$r_0 + \Pi_{r_0}^{-1}(r_2) + y \in \text{Img}(G)$	2.47	
	Verif. 2	$Ver(c_1, E(t_1), s_1)$	0.459	Verif. 2	$Ver(c_1, E(r_1), s_1)$	0.064	
		$Ver(c_2, E(t_2), s_2)$	0.446		$Ver(c_2, E(r_2), s_2)$	0.064	
		$w_H(t_1 + t_2)$	0.001		$w_R(r_1 + r_2)$	0.018	
		Total	349.037		Total	5.462	

Notable Observations



- The generation of the commitment is slower in the rank metric because the algorithm that generates an error of certain rank slow.
- The verification of the commitment is slower in the rank metric because computing the rank of a matrix is slower than computing the Hamming weight of a vector.
- The generation of matrix A (Hamming metric) is slower than G (Rank metric) because of their difference in the dimensions.
- Verification time of Zero-Knowledge proofs in the rank metric is around 70-100 times faster than the Hamming metric.

Conclusions and Future Work



Conclusions



- We showed that quantum-resistant commitments and zero-knowledge proofs can be built upon the Rank Syndrome Decoding problem.

Conclusions



- We showed that quantum-resistant commitments and zero-knowledge proofs can be built upon the Rank Syndrome Decoding problem.
- Our protocol is quasi-linear in the size of the circuit and has soundness $2/3$.

Conclusions



- We showed that quantum-resistant commitments and zero-knowledge proofs can be built upon the Rank Syndrome Decoding problem.
- Our protocol is quasi-linear in the size of the circuit and has soundness $2/3$.
- Provide implementations of both variants (Hamming and Rank) with parameters achieving 128 bits of security.

Future work



- Use structured codes (quasi-cyclic) to further improve efficiency and performance.
- Look for a better proof construction than iterative challenge response.
- Design and implementation of the 5-pass version.

References I



Gaborit, P., Schrek, J., and Zémor, G. (2011). Full cryptanalysis of the chen identification protocol. In Yang, B.-Y., editor, *Post-Quantum Cryptography*, pages 35–50, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jain, A., Krenn, S., Pietrzak, K., and Tentes, A. (2012). Commitments and efficient zero-knowledge proofs from learning parity with noise. In Wang, X. and Sako, K., editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 663–680, Berlin, Heidelberg. Springer Berlin Heidelberg.

