

The PyM Poly functions	
Function signature	Description
rd(K,s) rd(K,s,r) rd_nonzero(K,s) rd_nonzero(K,s,r)	Given a domain K and a integer s, rd returns a list of s-elements of K. In case of polynomials,another parameter r is needed to define the maximum degree of any possible polynomial. The rd_nonzero function return a list of s non zero elements of K.
rd_vector(K,n)	Given a domain K and a integer n, it returns a vector of n elements in K.
rd_linear_combination(G) rd_linear_combination(G,K)	If G is matrix and K is a field (or a ring), rd_linear_combination(G, K) returns a linear combination of the rows of G with coefficients randomly chosen in K.
combination(n,k)	Given integers k and n, it returns an ordered list of k different elements in the range (0..n-1)
permutation(n) rd_permutation(n)	Given an integer n, it return a permutation of the elements (0..n-1). If n is a list or vector, it returns a list with a permutation of the elements of n.
permutation_matrix(p) rd_permutation_matrix(n)	If p is a permutation of 0, 1, ..., n-1, this function creates the $n \times n$ matrix that has 0 everywhere, except at the entries (j,p[j]) that are set to 1. If p is an integer n, it agrees with rd_permutation_matrix(n) that selects p at random and returns permutation_matrix(p).
rd_nonzero_vector(K,n)	Given a domain K and a integer n, it returns a vector of n elements in K with not all elements equal to zero.
alternant_matrix (h, a, r) alternant_matrix (P ,A)	The call alternant_matrix (h, a, r) provides the alternant control matrix of order r associated to the vectors h and a. The call alternant_matrix(P,A), where $P=[p_1, \dots, p_r]$ is assumed to be a vector of univariate polynomials and $A=[a_1, \dots, a_n]$ a vector, constructs the matrix $(p_i(a_j))$.
scramble_matrix(A,k)	It returns a random $k \times k$ matrix M with elements in A with $ \det(M) = 1$
rd_error_vector (n,s,K) rd_error_vector (n,s) rd_error_pattern = rd_error_vector	If n and s are positive intgers, $s \leq n$, and K is a field (or a ring), rd_error_vector(n,s,K) produces a vector in K^n of weight s whose non-zero positions and the corresponding values have been chosen randomly. The function rd_error_vector(n, s) is defined as rd_error_vector(n,s, $\mathbb{Z}_n(2)$).
P_e(n,t,p)	It computes the probability that an event of probability p happens more than t times in n tries.
erf(n,t,p)	It computes $P_e(n,t,p)/p$. If p is 0, it returns 1.
rd_pick(P) rd_pick(P,k)	It returns a random element of the domain P. In case of polynomials, k gives the largest possible degree.
pick_element(k,A,variable) pick_element(k,A) element=pick=pick_element	It returns the k-th element of A from the $X_-(A)$. If the domain depends on a variable, the parameter var represent that variable
pick_nonzero_element(k,A,var) pick_nonzero_element(k,A) nonzero_element=pick_nonzero=pick_nonzero_element	It returns the k-th non zero element of A from the $X_-(A)$. If the domain depends on a variable, the parameter var represent that variable

The PyM Poly functions	
Function signature	Description
rd_selection(X,k=1) combination(n,k=1) permutation(n) Note: rd_selection = rd_choice = rd_sample	Returns a list of m elements of X (a list, tuple, vector, string). The indices of the elements in X are distinct. Therefore the returned elements are distinct if X has no repeated elements. Examples: rd_selection("abcdefghijklmnopqrstuvwxyz", 6) ⇒ ['j', 'r', 'f', 'm', 'e', 'c'] combination(10,5) ⇒ [2, 3, 4, 7, 8] permutation(10) ⇒ [5, 3, 7, 2, 8, 0, 1, 6, 9, 4]
rd_nonzero_choice(X,m=1)	
rd_matrix(K,m,n)	Delivers an m x n matrix with entries chosen randomly in K. For finite K, the sampling is uniform. For Z_ and Q_, the sampling distribution is bell-like around 0. For polynomials, the degree is chosen randomly in N_, and the coefficients randomly in the coefficient domain.
get_prime(d) secrets_set(d) shares(s,r,n)	Utility functions for the implementation of Shamir's secret sharing algorithms. get_prime(d) returns a random d-digit prime number.