# Package 'HEtools'

May 7, 2026

**Title** Homomorphic Encryption Polynomials

**Version** 1.0.0

**Description** Homomorphic encryption (Brakerski and Vaikuntanathan (2014) <doi:10.1137/120868669>) using Ring Learning with Errors (Lyubashevsky et al. (2012) <https://eprint.iacr.org/2012/230>) is a form of Learning with Errors (Regev (2005) <doi:10.1145/1060590.1060603>) using polynomial rings over finite fields. Functions to generate the required polynomials (using 'polynom'), with various distributions of coefficients are provided. Additionally, functions to generate and take coefficient modulo are provided.

**Depends** polynom

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Bastiaan Quast [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2951-3577>)

**Maintainer** Bastiaan Quast <bquast@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-12-08 13:30:02 UTC

## Contents

## CoefMod *Coefficient Modulo*

### Description

Coefficient Modulo

### Usage

```
CoefMod(x, k)
```

### Arguments

| | |
|---|---|
| x | polynomial from the polynom package |
| k | the modulo |

### Value

polynomial of the polynom class

### Examples

```
polynomial = polynomial(c(5, 3, 6))
print(polynomial)

CoefMod(polynomial, 5)
```

## GenDiscrGauss *Generate Polynomial with Discrete Gaussian Coefficients*

### Description

Generate Polynomial with Discrete Gaussian Coefficients

### Usage

```
GenDiscrGauss(n, s = 3)
```

### Arguments

| | |
|---|---|
| n | the order |
| s | scale the sigma (down) |

### Value

polynomial of the form x^^n + 1

## Examples

```
n = 5
GenDiscrGauss(n)

GenDiscrGauss(n=5, s=2)
```

---

GenPolyMod *Generate Polynomial Modulo*

---

## Description

Generate Polynomial Modulo

## Usage

```
GenPolyMod(n)
```

## Arguments

| | |
|---|---|
| n | the order |

## Value

polynomial of the form x^^n + 1

## Examples

```
n = 5
GenPolyMod(5)
```

---

GenTernary *Generate Polynomial with Ternary*

---

## Description

Generate Polynomial with Ternary

## Usage

```
GenTernary(n)
```

## Arguments

| | |
|---|---|
| n | the order |

## Value

ternary polynomial of order x^^n with coefficients (-1,0,1)

## Examples

```
n = 5
GenTernary(n)
```

---

GenUnif *Generate Polynomial with Uniform Distribution Coefficients*

---

## Description

Generate Polynomial with Uniform Distribution Coefficients

## Usage

```
GenUnif(n, q)
```

## Arguments

| | |
|---|---|
| n | the order |
| q | the ciphermod of coefficients |

## Value

polynomial of order x^^n with coefficients 0,..,q

## Examples

```
n = 5
q = 7
GenUnif(n, q)
```

# Index