Package 'HomomorphicEncryption'

October 21, 2022

Title BFV, BGV, CKKS Schema for Fully Homomorphic Encryption

Version 0.1.0

Description Implements the Brakerski-Fan-Vercauteren (BFV, 2012) <https://eprint.iacr.org/2012/144>, Brakerski-Gentry-Vaikuntanathan (BGV, 2014) <doi:10.1145/2633600>, and Cheon-Kim-Kim-Song (CKKS, 2016) <https://eprint.iacr.org/2016/421.pdf> schema for Fully Homomorphic Encryption, as well as several helper functions.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.1

Depends polynom, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Bastiaan Quast <bquast@gmail.com>

Repository CRAN

Date/Publication 2022-10-21 15:27:58 UTC

R topics documented:

`BFV_encrypt` *BFV encryption*

Description

BFV encryption

Usage

```
BFV_encrypt(m, pk)
```

Arguments

`m`	message
`pk`	public key

Value

polynomial

`BFV_KeyGen` *Brakerski / Fan-Vercauteren*

Description

Brakerski / Fan-Vercauteren

Usage

```
BFV_KeyGen()
```

Value

polynomial

Examples

```
d  =      4
n  =      2^d
p  =     (n/2)-1
q  = 424242
GenPolyMod(n)
```

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)
```

EncryptPoly0 *Encrypt Polynomial Message Part 0*

Description

Encrypt Polynomial Message Part 0

Usage

```
EncryptPoly0(m, pk0, u, e1, p, pm, q)
```

Arguments

m	message
pk0	public key part 0
u	u
e1	e1
p	p
pm	pm
q	q

Value

polynomial which contains the message in ciphertext

`EncryptPoly1` *Encrypt Polynomial Message Part 1*

Description

Encrypt Polynomial Message Part 1

Usage

```
EncryptPoly1(pk1, u, e2, pm, q)
```

Arguments

`pk1`	public key part 1
`u`	u
`e2`	e2
`pm`	pm
`q`	q

Value

polynomial which contains the message in ciphertext

`GenA` *Generate a*

Description

Generate a

Usage

```
GenA(n, q)
```

Arguments

`n`	the order
`q`	the ciphermod of coefficients

Value

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

Examples

```
n = 16
q =  7
GenA(n, q)
```

`GenError` *Generate a*

Description

Generate a

Usage

```
GenError(n)
```

Arguments

`n`	the order

Value

polynomial of order x^^n with discrete Gaussian distribution, bounded (not strictly true) by -n,n

Examples

```
n = 16
GenError(n)
```

`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 = 16
GenPolyMod(n)
```

GenPubKey *Generate the Public Key*

Description

Generate the Public Key

Usage

```
GenPubKey(a, n, e, pm)
```

Arguments

a	a
n	n
e	e
pm	pm

Value

list with the two polynomials that form the public key

GenPubKey0 *Generate part 0 of the Public Key*

Description

Generate part 0 of the Public Key

Usage

```
GenPubKey0(a, s, e, pm, q)
```

Arguments

a	a
s	s
e	e
pm	pm
q	q

Value

polynomial

`GenPubKey1` *Generate part 1 of the Public Key*

Description

Generate part 1 of the Public Key

Usage

```
GenPubKey1(a)
```

Arguments

`a`	a

Value

polynomial

`GenSecretkey` *Generate Secret key*

Description

Generate Secret key

Usage

```
GenSecretKey(n)
```

Arguments

`n`	the order

Value

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

Examples

```
n = 16
GenSecretKey(n)
```

GenU *Generate u*

Description

Generate u

Usage

```
GenU(n)
```

Arguments

n	the order

Value

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

Examples

```
n = 16
GenU
```

Index