Project 1.2
Revision 0

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 Class Documentation	3
2.1 Hill Class Reference	3
2.1.1 Detailed Description	3
2.1.2 Constructor & Destructor Documentation	3
2.1.2.1 Hill() [1/3]	3
2.1.2.2 Hill() [2/3]	4
2.1.2.3 Hill() [3/3]	5
2.1.3 Member Function Documentation	6
2.1.3.1 decrypt() [1/2]	6
2.1.3.2 decrypt() [2/2]	6
2.1.3.3 encrypt() [1/2]	7
2.1.3.4 encrypt() [2/2]	7
2.1.3.5 getD()	8
2.1.3.6 getE()	8
2.1.3.7 kpa()	8
2.1.3.8 setD()	9
2.1.3.9 setE()	9
Index	11

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:	
Hill	3

2 Class Index

Chapter 2

Class Documentation

2.1 Hill Class Reference

```
#include <Hill.hpp>
```

Public Member Functions

- Hill ()
- Hill (const Matrix &K, bool encryption)
- Hill (const Matrix &E, const Matrix &D)
- Matrix getE () const
- Matrix getD () const
- bool setE (const Matrix &E)
- bool setD (const Matrix &D)
- std::string encrypt (const std::string &P) const
- std::string encrypt (const std::string &P, const Matrix &E)
- std::string decrypt (const std::string &C) const
- std::string decrypt (const std::string &C, const Matrix &D)
- bool kpa (const std::vector< std::string > &P, const std::vector< std::string > &C, unsigned int n)

2.1.1 Detailed Description

A C++ class to perform encryption/decryption and cryptanalysis using/of the Hill cipher with a 29 character alphabet.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Hill() [1/3]

```
Hill::Hill ( )
```

Default constructor. It should set the encryption key to {2,4,3,5} (2-by-2) and the decryption key to its inverse.

2.1.2.2 Hill() [2/3]

Parameterized constructor. Use the parameter to set the encryption (E) and decryption (D) keys; if parameter is invalid then set E/D to a 0-by-0 matrix.

2.1 Hill Class Reference 5

Parameters

matrix represent- ing the en- cryp- tion or de- cryp- tion key. encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion key key is		
representing the encryption or decryption key. encryption - true if the key is the encryption key, false if the key is the decryption	K	- a
encryption encryption or decryption encryption - true if the key is the encryption key, false if the key is the decryption		matrix
ing the en- cryption or de- cryption key. encryption - true if the key is the en- cryption key, false if the key is the de- cryption cryption cryption cryption key, false if the key is the de- cryption cryption cryption		repre-
en- cryp- tion or de- cryp- tion key. encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion		sent-
cryption or decryption key. encryption if the key is the encryption key, false if the key is the decryption		ing the
tion or decryption key. encryption - true if the key is the encryption key, false if the key is the decryption		en-
encryption key. encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion		cryp-
encryption encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion		tion
encryption - true if the key is the encryption key, false if the key is the decryption		or de-
encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion		cryp-
encryption - true if the key is the en- cryp- tion key, false if the key is the de- cryp- tion		tion
if the key is the encryption key, false if the key is the decryption		key.
key is the encryption key, false if the key is the decryption	encryption	- true
the encryption key, false if the key is the decryption		if the
cryp- tion key, false if the key is the de- cryp- tion		key is
tion key, false if the key is the de- cryp- tion		the en-
key, false if the key is the de- cryp- tion		cryp-
false if the key is the de- cryp- tion		tion
if the key is the decryption		key,
key is the de- cryp- tion		false
the de- cryp- tion		if the
cryp- tion		key is
tion		the de-
		cryp-
key		tion
1		kov

2.1.2.3 Hill() [3/3]

```
Hill::Hill (  \mbox{const Matrix \& $E$,} \\ \mbox{const Matrix \& $D$ })
```

Parameterized constructor. Use the parameters to set the encryption (E) and decryption (D) keys; if a parameter is invalid or inconsistent then set E/D to a 0-by-0 matrix.

E	-	en-
	cr	ур-
	tic	n
	ke	ey.
D	-	de-
	or	\ (D
	CI	ур-
	tic	• •

2.1.3 Member Function Documentation

2.1.3.1 decrypt() [1/2]

```
std::string Hill::decrypt (  {\tt const \ std::string \ \& \ C \ ) \ const}
```

Decrypt the given ciphertext using the previous set decryption key, an empty string if the decryption key is invalid.

Parameters

```
C - the cipher-text to decrypt
```

Returns

the plaintext resulting from decrypting the ciphertext using the stored decryption matrix.

2.1.3.2 decrypt() [2/2]

Decrypt the given ciphertext using the given decryption key, an empty string if the decryption key is invalid.

С	- the
	plain-
	text
	to en-
	crypt
D	- the
	key to
	use
	to de-
	crypt
	the
	cipher-
	text

2.1 Hill Class Reference 7

Returns

the plaintext resulting from decrypting the ciphertext using the given decryption matrix.

2.1.3.3 encrypt() [1/2]

```
std::string Hill::encrypt (  {\tt const \ std::string \ \& \ P \ ) \ const}
```

Encrypt the given plaintext using the previous set encryption key, an empty string if the encryption key is invalid.

Parameters

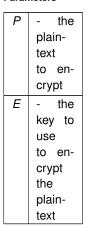
Р	- the
	plain-
	text
	to en-
	crypt

Returns

the ciphertext resulting from encrypting the plaintext using the stored encryption matrix.

2.1.3.4 encrypt() [2/2]

Encrypt the given plaintext using the given encryption key, an empty string if the encryption key is invalid.



Returns

the ciphertext resulting from encrypting the plaintext using the given encryption matrix.

2.1.3.5 getD()

```
Matrix Hill::getD ( ) const
```

Returns the current decryption key.

Returns

the decryption key (Matrix D), if no decryption key is set a 0-by-0 matrix.

2.1.3.6 getE()

```
Matrix Hill::getE ( ) const
```

Returns the current encryption key.

Returns

the encryption key (Matrix E), if no encryption key is set a 0-by-0 matrix.

2.1.3.7 kpa()

Mount a known-plaintext attack against the Hill cipher assuming an n-by-n encryption matrix. Set E/D to the encryption/decryption key if they can be recovered.

P	- the
	plain-
	texts
	that
	corre-
	spond
	to C
С	- the
	cipher-
	texts
	that
	corre-
	spond
	to P

2.1 Hill Class Reference 9

Returns

true if the encryption and decryption keys have been recovered.

2.1.3.8 setD()

```
bool Hill::setD ( {\tt const\ Matrix\ \&\ D\ )}
```

Sets the decryption key (Matrix D) and encryption key (Matrix E); if the parameter is invalid then set E/D to a 0-by-0 matrix.

Parameters



Returns

true if set is successful, false otherwise.

2.1.3.9 setE()

```
bool Hill::setE ( {\tt const\ Matrix\ \&\ E\ )}
```

Sets the encryption key (Matrix E) and decryption key (Matrix D); if the parameter is invalid then set E/D to a 0-by-0 matrix.

Parameters



Returns

true if set is successful, false otherwise.

The documentation for this class was generated from the following file:

· Hill.hpp

Index

```
decrypt
     Hill, 6
encrypt
     Hill, 7
getD
     Hill, 8
getE
     Hill, 8
Hill, 3
     decrypt, 6
     encrypt, 7
getD, 8
     getE, 8
     Hill, 3, 5
     kpa, 8
     setD, 9
     setE, 9
kpa
     Hill, 8
setD
     Hill, 9
setE
     Hill, 9
```