GenericCryptanalyzer

Generated by Doxygen 1.9.1

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

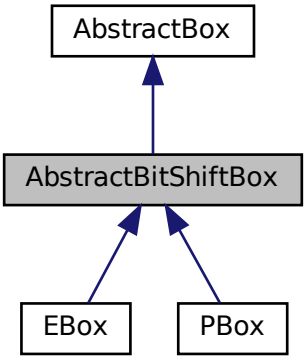
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

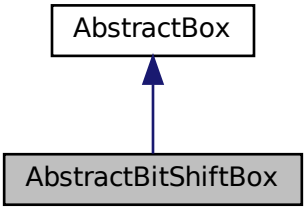
Class Documentation

3.1 AbstractBitShiftBox Class Reference

Inheritance diagram for AbstractBitShiftBox:



Collaboration diagram for AbstractBitShiftBox:



Public Member Functions

- **AbstractBitShiftBox** (size_t in_size, size_t out_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes, const vector< size_t > &bit_src)
- **AbstractBitShiftBox** (size_t in_size, size_t out_size, const vector< size_t > &bit_src)
- void determine_next () override

  determine_next method to determine the next best output sorted by probabilities

Protected Member Functions

- void **apply_transformation** ()

Protected Attributes

- vector< size_t > **bit_src**

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

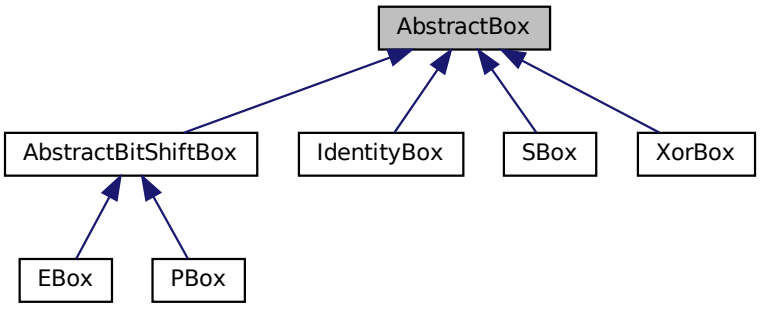
- src/box/abstractbitshiftbox.h
- src/box/abstractbitshiftbox.cpp

3.2 AbstractBox Class Reference

An AbstractBox represents an abstract idea of a block cipher component such as a Pbox, Sbox, Xor, Addition, etc. A cipher is composed of multiple such `boxes` that communicate with each other through connections.

`#include <abstractbox.h>`

Inheritance diagram for AbstractBox:



Public Member Functions

- AbstractBox (size_t in_size, size_t out_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes)

  AbstractBox.
- AbstractBox (size_t in_size, size_t out_size)

  AbstractBox similar to the previous constructor, but leaves dst_boxes empty.
- void add_dest (AbstractBoxPtr dst_box, BitsRange out_range, BitsRange in_range)

  add_dest adds a new destination box for the output of the box to flow to
- const dynamic_bitset & get_input ()

  get_input getter for `in_bits`
- const dynamic_bitset & get_output ()

  get_output getter for `out_bits`
- size_t input_size ()

  input_size getter for the size of `in_bits`
- size_t output_size ()

  output_size getter for the size of `out_bits`
- bool is_determined ()

  is_determined getter for `is_det`
- virtual void set_input (dynamic_bitset<> bits, const BitsRange &rng)

  set_input sets a subrange `rng` *of the input to the value of* `bits`
- void notify_all ()

  notify_all notifies all the destination boxes after the output of the box is determined
- virtual void determine_next ()=0

  determine_next method to determine the next best output sorted by probabilities
- virtual void reset_determination ()

  reset_determination set the process to be undetermined by setting `is_det` *to* `false`
- double get_probability ()

  get_probability get the probability of the current differential characteristic

Protected Attributes

- dynamic_bitset in_bits

  in_bits the bits that flow into the box
- dynamic_bitset out_bits

  out_bits the bits that flow out of the box
- vector< pair< AbstractBoxPtr, Connection > > dst_boxes

  dst_boxes describes how the `out_bits` *flow from this box to other following boxes*
- bool is_det

  is_det a boolean value that should be true if and only if at least one out of all possible outputs has been determined and returned
- double prob

  prob the probability of the box to output the currently determined state

Friends

- class **RoundFunction**

3.2.1 Detailed Description

An AbstractBox represents an abstract idea of a block cipher component such as a Pbox, Sbox, Xor, Addition, etc. A cipher is composed of multiple such `boxes` that communicate with each other through connections.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 AbstractBox() [1/2]

```
AbstractBox::AbstractBox (
            size_t in_size,
            size_t out_size,
            const vector< pair< AbstractBoxPtr, Connection >> & dst_boxes )
```

AbstractBox.

Parameters

in_size	size of the input bits of this box
out_size	size of the output bits of this box
dst_boxes	output flow connections to following boxes

3.2.2.2 AbstractBox() [2/2]

```
AbstractBox::AbstractBox (
            size_t in_size,
            size_t out_size )
```

AbstractBox similar to the previous constructor, but leaves dst_boxes empty.

Parameters

in_size	size of the input bits of this box
out_size	size of the output bits of this box

3.2.3 Member Function Documentation

3.2.3.1 add_dest()

```
void AbstractBox::add_dest (
            AbstractBoxPtr dst_box,
```

```
            BitsRange out_range,
            BitsRange in_range )
```

add_dest adds a new destination box for the output of the box to flow to

Parameters

dst_box	a pointer to the destination box
out_range	a subrange of `out_bits` from this box that will flow to `dst_box`
in_range	a subrange of `in_bit` from `dst_box` into which the bits will flow

3.2.3.2 get_input()

```
const dynamic_bitset & AbstractBox::get_input ( )
```

get_input getter for `in_bits`

Returns

`in_bits`

3.2.3.3 get_output()

```
const dynamic_bitset & AbstractBox::get_output ( )
```

get_output getter for `out_bits`

Returns

`out_bits`

3.2.3.4 get_probability()

```
double AbstractBox::get_probability ( )
```

get_probability get the probability of the current differential characteristic

Returns

`prob`

3.2.3.5 input_size()

```
size_t AbstractBox::input_size ( )
```

input_size getter for the size of `in_bits`

Returns

`in_bits.size()`

3.2.3.6 is_determined()

```
bool AbstractBox::is_determined ( )
```

is_determined getter for `is_det`

Returns

`is_det`

3.2.3.7 output_size()

```
size_t AbstractBox::output_size ( )
```

output_size getter for the size of `out_bits`

Returns

`out_bits.size()`

3.2.3.8 set_input()

```
void AbstractBox::set_input (
            dynamic_bitset<> bits,
            const BitsRange & rng )  [virtual]
```

set_input sets a subrange `rng` of the input to the value of `bits`

Parameters

bits	the bits that will be put in `in_bits`
rng	the subrange in which bits will be put in `in_bits`

Reimplemented in SBox.

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

- src/box/abstractbox.h
- src/box/abstractbox.cpp

3.3 BitsRange Struct Reference

Public Attributes

- size_t **start**
- size_t **len**

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

- src/helpers/helpers.h

3.4 CipherAnalyzer Class Reference

Public Member Functions

- **CipherAnalyzer** (vector< RoundFunctionPtr > rounds, size_t input_max_hamming_weight, double global↩ _thresh, vector< double > opt_probs)
- **CipherAnalyzer** (vector< RoundFunctionPtr > rounds, size_t input_max_hamming_weight, double global↩ _thresh)
- ProbEntry **get_next_entry** ()
- void **set_input** (const dynamic_bitset<> &bits, BitsRange rng)

Protected Member Functions

- bool **advance_state** ()

Protected Attributes

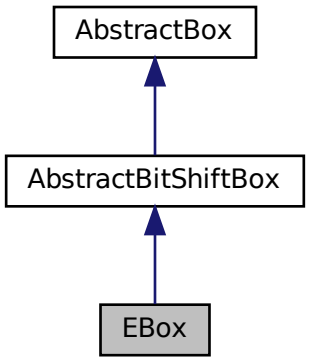
- double **global_thresh**
- vector< double > **opt_probs**
- vector< double > **round_probs**
- vector< RoundFunctionPtr > **rounds**
- size_t **curr_idx**

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

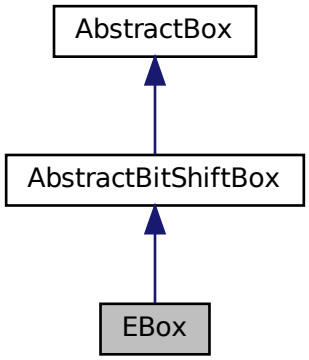
- src/cipheranalyzer.h
- src/cipheranalyzer.cpp

3.5 EBox Class Reference

Inheritance diagram for EBox:



Collaboration diagram for EBox:



Public Member Functions

- **EBox** (size_t in_size, size_t out_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes, const vector< size_t > &bit_expansion)
- **EBox** (size_t in_size, size_t out_size, const vector< size_t > &bit_expansion)

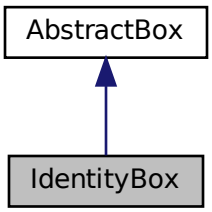
Additional Inherited Members

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

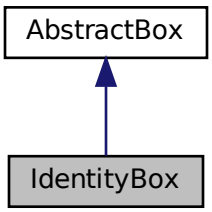
- src/box/ebox.h
- src/box/ebox.cpp

3.6 IdentityBox Class Reference

Inheritance diagram for IdentityBox:



Collaboration diagram for IdentityBox:



Public Member Functions

- **IdentityBox** (size_t data_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes)
- **IdentityBox** (size_t data_size)
- void determine_next () override

  determine_next method to determine the next best output sorted by probabilities

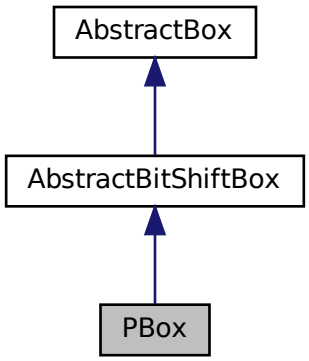
Additional Inherited Members

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

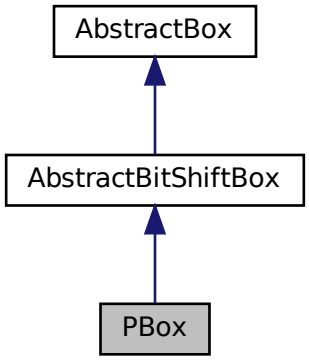
- src/box/identitybox.h
- src/box/identitybox.cpp

3.7 PBox Class Reference

Inheritance diagram for PBox:



Collaboration diagram for PBox:



Public Member Functions

- **PBox** (size_t bits_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes, const vector< size_t > &bit_perm)
- **PBox** (size_t bits_size, const vector< size_t > &bit_perm)

Additional Inherited Members

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

- src/box/pbox.h
- src/box/pbox.cpp

3.8 RoundFunction Class Reference

Public Member Functions

- **RoundFunction** (string src_id, string dst_id, map< string, AbstractBoxConstructor > constrs, map< string, vector< NamedConnection >> conns)
- bool **is_determined** ()
- ProbEntry **get_next_entry** ()
- void **set_input** (const dynamic_bitset<> bits, BitsRange rng)
- void **set_threshold** (double beta)

Protected Member Functions

- bool **advance_state** ()
- void **top_sort_boxes** (AbstractBoxPtr src, vector< AbstractBoxPtr > &top_sort, map< AbstractBoxPtr, bool > &is_visited)
- vector< AbstractBoxPtr > **sort_boxes** (AbstractBoxPtr src)

Protected Attributes

- AbstractBoxPtr **src**
- AbstractBoxPtr **dst**
- size_t **curr_box_idx**
- vector< AbstractBoxPtr > **boxes**
- vector< double > **partial_prob**
- double **beta_thresh**
- bool **is_det**

Friends

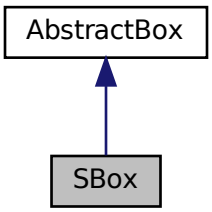
- class **CipherAnalyzer**

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

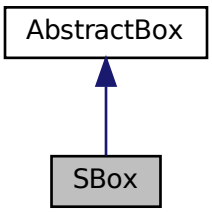
- src/roundfunction.h
- src/roundfunction.cpp

3.9 SBox Class Reference

Inheritance diagram for SBox:



Collaboration diagram for SBox:



Public Member Functions

- **SBox** (size_t in_size, size_t out_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes, const ProbTable &prob_table)
- **SBox** (size_t in_size, size_t out_size, const ProbTable &prob_table)
- void determine_next () override

  determine_next method to determine the next best output sorted by probabilities
- void reset_determination () override

  reset_determination set the process to be undetermined by setting `is_det` *to* `false`
- void set_input (dynamic_bitset<> bits, const BitsRange &rng) override

  set_input sets a subrange `rng` *of the input to the value of* `bits`

Protected Member Functions

- size_t **convert_to_index** (const dynamic_bitset<> &bits)

Protected Attributes

- ProbTable **prob_table**
- size_t **table_idx**
- size_t **table_entry**

3.9.1 Member Function Documentation

3.9.1.1 set_input()

```
void SBox::set_input (
            dynamic_bitset<> bits,
            const BitsRange & rng )  [override], [virtual]
```

set_input sets a subrange `rng` of the input to the value of `bits`

Parameters

bits	the bits that will be put in `in_bits`
rng	the subrange in which bits will be put in `in_bits`

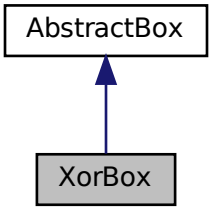
Reimplemented from AbstractBox.

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

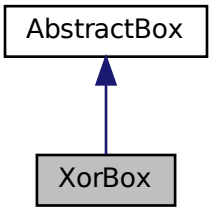
- src/box/sbox.h
- src/box/sbox.cpp

3.10 XorBox Class Reference

Inheritance diagram for XorBox:



Collaboration diagram for XorBox:



Public Member Functions

- **XorBox** (size_t data_size, const vector< pair< AbstractBoxPtr, Connection >> &dst_boxes)
- **XorBox** (size_t data_size)
- void determine_next () override

  determine_next method to determine the next best output sorted by probabilities

Additional Inherited Members

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

- src/box/xorbox.h
- src/box/xorbox.cpp

Index