Searchable_HEDB

1

Generated by Doxygen 1.8.13

Contents

Chapter 1

HOW TO RUN

Searchable DB Library Install

```
cd ./big3_searchable_hedb/HDB_comparison_library
rm -r build
mkdir build
cd build
```

If HElib is installed as a library...

```
cmake ..
```

If HElib is locally installed...

```
cmake -Dhelib_DIR=/{PATH}/helib_install/helib_pack/share/cmake/helib ..
```

then

```
make install
```

installs the HDB library in the folder ./big3_searchable_hedb/lib_HDB

To run main code...

```
cd ./big3_searchable_hedb
rm -r build
mkdir build
cd build
```

then run cmake and make as above. Compiled binary will be in ./big3_searchable_hedb/bin

API

Can be found in the ./html directory. Open index.html to access the API documentation.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 HDB_supergate_ Namespace Reference

Classes

- struct BGV_param
- class CSVIterator
- class CSVRange
- class CSVRow
- class CtxtIndex

  class representing an encrypted ciphertext index
- class CtxtIndexFile

  class representing a ciphertext index file
- class HEQuery

  Object representing a query object used to query the HEDB.
- class PtxtIndex

  class representing a plaintext index
- class PtxtIndexFile

  class representing a collection of PtxtIndexes
- struct STD128_HDB
- struct TOY_HDB

Typedefs

- typedef std::vector< helib::Ctxt > Ctxt_vec
- typedef std::vector< std::vector< helib::Ctxt > > Ctxt_mat

Enumerations

- enum Q_TYPE_t {
  EQ, **LT**, **LEQ**, **MIN**,
  MAX, **EQ**, **LT**, **LEQ**,
  MIN, **MAX** }
- enum Q_TYPE_t {
  EQ, **LT**, **LEQ**, **MIN**,
  MAX, **EQ**, **LT**, **LEQ**,
  MIN, **MAX** }

Functions

- std::istream & **operator>>** (std::istream &str, CSVRow &data)
- struct BGV_param MakeBGVParam (long, long, long, long, long, long, long, long)
- helib::Context **MakeBGVContext** (long, long, long, long, long, long)
- helib::Context MakeBGVContext (const struct BGV_param)
- void **setIndexParams** (unsigned long, unsigned long, unsigned long, unsigned long &, unsigned long &, bool)
- void **dataToZZXSlot** (unsigned long data, vector< ZZX > &dest, unsigned long counter, unsigned long digit_base, unsigned long exp_len, unsigned long enc_base, he_cmp::Comparator &comparator)
- void **encryptAndInsert** (const helib::Context &contx, helib::PubKey &pk, std::vector< NTL::ZZX > &ptxt, Ctxt_vec &dest)
- long **findNSlots** (long, long)
- istream & **operator>>** (istream &str, CSVRow &data)
- void **encryptAndInsert** (const Context &contx, PubKey &pk, vector< ZZX > &ptxt, Ctxt_vec &dest)

4.1.1 Detailed Description

main namespace for all utility functions for a HEDB

4.1.2 Typedef Documentation

4.1.2.1 Ctxt_mat

```
typedef std::vector< std::vector< helib::Ctxt > > HDB_supergate_::Ctxt_mat
```

a matrix of ciphertexts, Ctxt_mat

4.1.2.2 Ctxt_vec

```
typedef std::vector< helib::Ctxt > HDB_supergate_::Ctxt_vec
```

a vector of ciphertexts, Ctxt_vec

4.1.3 Enumeration Type Documentation

4.1.3.1 Q_TYPE_t [1/2]

```
enum HDB_supergate_::Q_TYPE_t
```

Query Type Enum A query can be equal EQ, less than LT, or less than or equal to LEQ. MIN and MAX queries are not supported yet.

4.1.3.2 Q_TYPE_t [2/2]

```
enum HDB_supergate_::Q_TYPE_t
```

Query Type Enum A query can be equal EQ, less than LT, or less than or equal to LEQ. MIN and MAX queries are not supported yet.

4.1.4 Function Documentation

4.1.4.1 MakeBGVContext()

```
helib::Context HDB_supergate_::MakeBGVContext (
             const struct BGV_param  )
```

function to create a helib::Context given parameters

4.1.4.2 MakeBGVParam()

```
struct BGV_param HDB_supergate_::MakeBGVParam (
             long ,
             long ,
             long ,
             long ,
             long ,
             long ,
             long ,
             long  )
```

function to create BGV_Param given parameters

4.2 HDB_supergate_server_ Namespace Reference

Namespace for the SERVER class.

Classes

- class SERVER

  Class that contains the DB and is queried upon.

4.2.1 Detailed Description

Namespace for the SERVER class.

4.3 HDB_supergate_user_ Namespace Reference

Namespace for the USER class.

Classes

- class USER
  Class that simulates the USER that queries the DB.

4.3.1 Detailed Description

Namespace for the USER class.

Chapter 5

Class Documentation

5.1 HDB_supergate_::BGV_param Struct Reference

```
#include <HDB_supergate.hpp>
```

Public Attributes

- long **p**
- long **d**
- long **m**
- long **nb_primes**
- long **expansion_len**
- long **c**
- long **scale**
- long **r**

5.1.1 Detailed Description

BGV_param struct A struct representing all necessary parameters to construct a BGV crypto context and the necessary comparison logic.

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.2 he_cmp::Comparator Class Reference

Public Member Functions

- DoubleCRT **create_shift_mask** (double &size, long shift)
- void **create_all_shift_masks** ()
- void **compute_poly_params** ()
- void **create_poly** ()
- void **extraction_init** ()
- void **extract_mod_p** (vector< Ctxt > &mod_p_coefs, const Ctxt &ctxt_x) const
- void **batch_shift** (Ctxt &ctxt, long start, long shift) const
- void **batch_shift_for_mul** (Ctxt &ctxt, long start, long shift) const
- void **shift_and_add** (Ctxt &x, long start, long shift_direction=false) const
- void **shift_and_mul** (Ctxt &x, long start, long shift_direction=false) const
- void **mapTo01_subfield** (Ctxt &ctxt, long pow) const
- void **evaluate_univar_less_poly** (Ctxt &ret, Ctxt &ctxt_p_1, const Ctxt &x) const
- void **evaluate_min_max_poly** (Ctxt &ctxt_min, Ctxt &ctxt_max, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_bivar** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_bivar_tan** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_mod_2** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_mod_3** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_mod_5** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_mod_7** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **less_than_mod_any** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **is_zero** (Ctxt &ctxt_res, const Ctxt &ctxt_z, long pow=1) const
- void **min_max_digit** (Ctxt &ctxt_min, Ctxt &ctxt_max, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **int_to_slot** (ZZX &poly, unsigned long input, unsigned long enc_base) const
- void **get_sorting_index** (vector< Ctxt > &ctxt_out, const vector< Ctxt > &ctxt_in) const
- void **find_prim_root** (ZZ_pE &root) const
- **Comparator** (const Context &context, CircuitType type, unsigned long d, unsigned long expansion_len, PubKey &pk, bool verbose)
- const DoubleCRT & **get_mask** (double &size, long index) const
- const ZZX & **get_less_than_poly** () const
- const ZZX & **get_min_max_poly** () const
- void **compare** (Ctxt &ctxt_res, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **min_max** (Ctxt &ctxt_min, Ctxt &ctxt_max, const Ctxt &ctxt_x, const Ctxt &ctxt_y) const
- void **array_min** (Ctxt &ctxt_res, const vector< Ctxt > &ctxt_in, long depth=0) const
- void **sort** (vector< Ctxt > &ctxt_out, const vector< Ctxt > &ctxt_in) const

Public Attributes

- const Context & **m_context**
- unsigned long **m_slotDeg**
- unsigned long **m_expansionLen**
- vector< DoubleCRT > **m_mulMasks**
- vector< double > **m_mulMasksSize**
- CircuitType **m_type**
- ZZX **m_univar_less_poly**
- ZZX **m_univar_min_max_poly**
- mat_ZZ **m_bivar_less_coefs**
- long **m_bs_num_comp**
- long **m_bs_num_min**
- long **m_gs_num_comp**

- long **m_gs_num_min**
- ZZ **m_top_coef_comp**
- ZZ **m_top_coef_min**
- ZZ **m_extra_coef_comp**
- ZZ **m_extra_coef_min**
- long **m_baby_index**
- long **m_giant_index**
- ZZX **m_slot_gen**
- PubKey & **m_pk**
- vector< vector< DoubleCRT > > **m_extraction_const**
- vector< vector< double > > **m_extraction_const_size**
- bool **m_verbose**

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

- HDB_comparison_library/comp_lib/comparator.h
- HDB_comparison_library/comp_lib/comparator.cpp

5.3 HDB_supergate_::CSVIterator Class Reference

Public Types

- typedef std::input_iterator_tag **iterator_category**
- typedef CSVRow **value_type**
- typedef std::size_t **difference_type**
- typedef CSVRow ∗ **pointer**
- typedef CSVRow & **reference**
- typedef std::input_iterator_tag **iterator_category**
- typedef CSVRow **value_type**
- typedef std::size_t **difference_type**
- typedef CSVRow ∗ **pointer**
- typedef CSVRow & **reference**

Public Member Functions

- **CSVIterator** (std::istream &str)
- CSVIterator & **operator++** ()
- CSVIterator **operator++** (int)
- CSVRow const & **operator**∗ () const
- CSVRow const ∗ **operator->** () const
- bool **operator==** (CSVIterator const &rhs)
- bool **operator!=** (CSVIterator const &rhs)
- **CSVIterator** (std::istream &str)
- CSVIterator & **operator++** ()
- CSVIterator **operator++** (int)
- CSVRow const & **operator**∗ () const
- CSVRow const ∗ **operator->** () const
- bool **operator==** (CSVIterator const &rhs)
- bool **operator!=** (CSVIterator const &rhs)

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.4 HDB_supergate_::CSVRange Class Reference

Public Member Functions

- **CSVRange** (std::istream &str)
- CSVIterator **begin** () const
- CSVIterator **end** () const
- **CSVRange** (std::istream &str)
- CSVIterator **begin** () const
- CSVIterator **end** () const

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.5 HDB_supergate_::CSVRow Class Reference

Public Member Functions

- std::string_view **operator[ ]** (std::size_t index) const
- std::size_t **size** () const
- void **readNextRow** (std::istream &str)
- std::string_view **operator[ ]** (std::size_t index) const
- std::size_t **size** () const
- void **readNextRow** (std::istream &str)

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.6 HDB_supergate_::CtxtIndex Class Reference

class representing an encrypted ciphertext index

```
#include <HDB_supergate.hpp>
```

Public Member Functions

- void **encrypt** (PtxtIndex ptIndex, he_cmp::Comparator &comparator, const helib::Context &contx, helib::↩ PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- Ctxt_vec & keys ()
- Ctxt_mat & uids ()
- unsigned long getX ()
- unsigned long getY ()
- void **encrypt** (PtxtIndex ptIndex, he_cmp::Comparator &comparator, const helib::Context &contx, helib::↩ PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- Ctxt_vec & keys ()
- Ctxt_mat & uids ()
- unsigned long getX ()
- unsigned long getY ()

5.6.1 Detailed Description

class representing an encrypted ciphertext index

A ciphertext index has a Ctxt_vec type of encrypted keys and Ctxt_mat type of encrypted uids. The encrypted uids has dimension of X rows and Y columns.

5.6.2 Member Function Documentation

5.6.2.1 getX() [1/2]

```
unsigned long HDB_supergate_::CtxtIndex::getX ( )  [inline]
```

getter for X value

5.6.2.2 getX() [2/2]

```
unsigned long HDB_supergate_::CtxtIndex::getX ( )  [inline]
```

getter for X value

5.6.2.3 getY() [1/2]

```
unsigned long HDB_supergate_::CtxtIndex::getY ( )  [inline]
```

getter for Y value

5.6.2.4 getY() [2/2]

```
unsigned long HDB_supergate_::CtxtIndex::getY ( )  [inline]
```

getter for Y value

5.6.2.5 keys() [1/2]

```
Ctxt_vec& HDB_supergate_::CtxtIndex::keys ( )  [inline]
```

getter for enc_key

5.6.2.6 keys() [2/2]

```
Ctxt_vec& HDB_supergate_::CtxtIndex::keys ( )  [inline]
```

getter for enc_key

5.6.2.7 uids() [1/2]

```
Ctxt_mat& HDB_supergate_::CtxtIndex::uids ( )  [inline]
```

getter for enc_uid

5.6.2.8 uids() [2/2]

```
Ctxt_mat& HDB_supergate_::CtxtIndex::uids ( )  [inline]
```

getter for enc_uid

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.7 HDB_supergate_::CtxtIndexFile Class Reference

class representing a ciphertext index file

```
#include <HDB_supergate.hpp>
```

Public Member Functions

- void **encrypt** (PtxtIndexFile &ptIndexFile, he_cmp::Comparator &comparator, const helib::Context &contx, helib::PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- void **insert** (std::string colname, PtxtIndex &ptIndex, he_cmp::Comparator &comparator, const helib::Context &contx, helib::PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- void insert (std::string colname, CtxtIndex &index)
- CtxtIndex & find (unsigned long)
- CtxtIndex & find (std::string)
- unsigned long indexOf (std::string)
- void **encrypt** (PtxtIndexFile &ptIndexFile, he_cmp::Comparator &comparator, const helib::Context &contx, helib::PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- void **insert** (std::string colname, PtxtIndex &ptIndex, he_cmp::Comparator &comparator, const helib::Context &contx, helib::PubKey &pk, unsigned long input_range, unsigned long digit_base, unsigned long enc_base, unsigned long exp_len, unsigned long nslots, unsigned long max_per, bool verbose)
- void insert (std::string colname, CtxtIndex &index)
- CtxtIndex & find (unsigned long)
- CtxtIndex & find (std::string)
- unsigned long indexOf (std::string)

5.7.1 Detailed Description

class representing a ciphertext index file

A ciphertext intex file is a collection of CtxtIndexes. It is a collection of <column name, CtxtIndex> pairs, with the column name and corresponding ciphertext index as a pair.

5.7.2 Member Function Documentation

5.7.2.1 find() [1/4]

```
CtxtIndex & HDB_supergate_::CtxtIndexFile::find (
             unsigned long i )
```

Finds the corresponding CtxtIndex given index of column

5.7.2.2 find() [2/4]

```
CtxtIndex& HDB_supergate_::CtxtIndexFile::find (
             unsigned long )
```

Finds the corresponding CtxtIndex given index of column

5.7.2.3 find() [3/4]

```
CtxtIndex& HDB_supergate_::CtxtIndexFile::find (
             std::string  )
```

Finds the corresponding CtxtIndex given column name

5.7.2.4 find() [4/4]

```
CtxtIndex& HDB_supergate_::CtxtIndexFile::find (
             std::string  )
```

Finds the corresponding CtxtIndex given column name

5.7.2.5 indexOf() [1/2]

```
unsigned long HDB_supergate_::CtxtIndexFile::indexOf (
             std::string  )
```

Returns the index given the column name

5.7.2.6 indexOf() [2/2]

```
unsigned long HDB_supergate_::CtxtIndexFile::indexOf (
             std::string  )
```

Returns the index given the column name

5.7.2.7 insert() [1/2]

```
void HDB_supergate_::CtxtIndexFile::insert (
            std::string colname,
            CtxtIndex & index )
```

Inserts CtxtIndex for given colname

5.7.2.8 insert() [2/2]

```
void HDB_supergate_::CtxtIndexFile::insert (
            std::string colname,
            CtxtIndex & index )
```

Inserts CtxtIndex for given colname

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

- HDB_comparison_library/include/HDB_supergate.hpp
- HDB_comparison_library/src/HDB_supergate.cpp

5.8 HDB_supergate_::HEQuery Class Reference

Object representing a query object used to query the HEDB.

```
#include <HDB_supergate.hpp>
```

Public Member Functions

- HEQuery (helib::PubKey &pk)
- void **insert** (unsigned long src, helib::Ctxt &EQ, helib::Ctxt &LT, helib::Ctxt &qry, std::vector< unsigned long > dst)
- HEQuery (helib::PubKey &pk)
- void **insert** (unsigned long src, helib::Ctxt &EQ, helib::Ctxt &LT, helib::Ctxt &qry, std::vector< unsigned long > dst)

Public Attributes

- unsigned long source
- helib::Ctxt query
- std::pair< helib::Ctxt, helib::Ctxt > Q_type
- std::vector< unsigned long > dest

5.8.1 Detailed Description

Object representing a query object used to query the HEDB.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 HEQuery() [1/2]

```
HDB_supergate_::HEQuery::HEQuery (
            helib::PubKey & pk ) [inline]
```

Constructor of the HEQuery class

The constructor takes in the public key to initialize query ciphertext and query type ctxt pair.

Parameters

pk	reference to the public key

5.8.2.2 HEQuery() [2/2]

```
HDB_supergate_::HEQuery::HEQuery (
             helib::PubKey & pk )  [inline]
```

Constructor of the HEQuery class

The constructor takes in the public key to initialize query ciphertext and query type ctxt pair.

Parameters

pk	reference to the public key

5.8.3 Member Data Documentation

5.8.3.1 dest

```
std::vector< unsigned long > HDB_supergate_::HEQuery::dest
```

Collection of destination columns to query. TODO: encrypt these

5.8.3.2 Q_type

```
std::pair< helib::Ctxt, helib::Ctxt > HDB_supergate_::HEQuery::Q_type
```

query type EQ <E(1), E(0)>, LT <E(0), E(1)>, or LEQ <E(1),E(1)>

5.8.3.3 query

```
helib::Ctxt HDB_supergate_::HEQuery::query
```

the query ciphertext

5.8.3.4 source

```
unsigned long HDB_supergate_::HEQuery::source
```

The source column index. TODO: encrypt this too

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.9 HDB_supergate_::PtxtIndex Class Reference

class representing a plaintext index

```
#include <HDB_supergate.hpp>
```

Public Member Functions

- void insert (long k, unsigned long v)
- int R ()
- int C ()
- std::vector< long > getKeys ()
- bool empty (long)
- long getSize (long)
- long popBack (long, bool emty=false)
- void printIndex ()
- void insert (long k, unsigned long v)
- int R ()
- int C ()
- std::vector< long > getKeys ()
- bool empty (long)
- long getSize (long)
- long popBack (long, bool emty=false)
- void printIndex ()

5.9.1 Detailed Description

class representing a plaintext index

A plaintext index is a collection of <key, [uid]> pairs, so each key is mapped to a list of uids that describe rows in the DB. Key are currently represented as integers.

5.9.2 Member Function Documentation

5.9.2.1 C() [1/2]

```
int HDB_supergate_::PtxtIndex::C ( )  [inline]
```

returns c, the maximum length of values array

5.9.2.2 C() [2/2]

```
int HDB_supergate_::PtxtIndex::C ( )  [inline]
```

returns c, the maximum length of values array

5.9.2.3 empty() [1/2]

```
bool HDB_supergate_::PtxtIndex::empty (
             long key )
```

true if queried key does not have any values mapped to it

5.9.2.4 empty() [2/2]

```
bool HDB_supergate_::PtxtIndex::empty (
             long  )
```

true if queried key does not have any values mapped to it

5.9.2.5 getKeys() [1/2]

```
std::vector<long> HDB_supergate_::PtxtIndex::getKeys ( )  [inline]
```

returns the keys vector

5.9.2.6 getKeys() [2/2]

```
std::vector<long> HDB_supergate_::PtxtIndex::getKeys ( )  [inline]
```

returns the keys vector

5.9.2.7 getSize() [1/2]

```
long HDB_supergate_::PtxtIndex::getSize (
             long key )
```

gets the size of index vector for given key

5.9.2.8 getSize() [2/2]

```
long HDB_supergate_::PtxtIndex::getSize (
             long  )
```

gets the size of index vector for given key

5.9.2.9 insert() [1/2]

```
void HDB_supergate_::PtxtIndex::insert (
             long k,
             unsigned long v )
```

inserts value v into key k

5.9.2.10 insert() [2/2]

```
void HDB_supergate_::PtxtIndex::insert (
             long k,
             unsigned long v )
```

inserts value v into key k

5.9.2.11 popBack() [1/2]

```
long HDB_supergate_::PtxtIndex::popBack (
             long key,
             bool emty = false )
```

removes the right-most key value from keys vector

5.9.2.12 popBack() [2/2]

```
long HDB_supergate_::PtxtIndex::popBack (
             long ,
             bool emty = false )
```

removes the right-most key value from keys vector

5.9.2.13 printIndex() [1/2]

```
void HDB_supergate_::PtxtIndex::printIndex ( )
```

debug function to print the index

5.9.2.14 printIndex() [2/2]

```
void HDB_supergate_::PtxtIndex::printIndex ( )
```

debug function to print the index

5.9.2.15 R() [1/2]

```
int HDB_supergate_::PtxtIndex::R ( )  [inline]
```

returns the number of keys

5.9.2.16 R() [2/2]

```
int HDB_supergate_::PtxtIndex::R ( )  [inline]
```

returns the number of keys

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

- HDB_comparison_library/include/HDB_supergate.hpp
- HDB_comparison_library/src/HDB_supergate.cpp

5.10 HDB_supergate_::PtxtIndexFile Class Reference

class representing a collection of PtxtIndexes

```
#include <HDB_supergate.hpp>
```

Public Member Functions

- std::vector< std::pair< std::string, PtxtIndex > > & getIndexFile ()
- void **insert** (std::string col, long k, unsigned long v)
- void printIndex (std::string col)
- void printIndexFile ()
- std::vector< std::pair< std::string, PtxtIndex > > & getIndexFile ()
- void **insert** (std::string col, long k, unsigned long v)
- void printIndex (std::string col)
- void printIndexFile ()

5.10.1 Detailed Description

class representing a collection of PtxtIndexes

The plaintext index file is the object representing lots of plaintext indexes. It is represented as a vector of <column name, PtxtIndex> pairs, with an index associated with each column of a DB.

5.10.2 Member Function Documentation

5.10.2.1 getIndexFile() [1/2]

```
std::vector<std::pair<std::string, PtxtIndex> >& HDB_supergate_::PtxtIndexFile::getIndexFile
( ) [inline]
```

getter for the IndexFile

5.10.2.2 getIndexFile() [2/2]

```
std::vector<std::pair<std::string, PtxtIndex> >& HDB_supergate_::PtxtIndexFile::getIndexFile
( ) [inline]
```

getter for the IndexFile

5.10.2.3 printIndex() [1/2]

```
void HDB_supergate_::PtxtIndexFile::printIndex (
             std::string col )
```

inserts for column col, key k, value v debug function for printing a particular index, given column name

5.10.2.4 printIndex() [2/2]

```
void HDB_supergate_::PtxtIndexFile::printIndex (
             std::string col )
```

inserts for column col, key k, value v debug function for printing a particular index, given column name

5.10.2.5 printIndexFile() [1/2]

```
void HDB_supergate_::PtxtIndexFile::printIndexFile ( )
```

debug function for printing the entire PtxtIndexFile

5.10.2.6 printIndexFile() [2/2]

```
void HDB_supergate_::PtxtIndexFile::printIndexFile ( )
```

debug function for printing the entire PtxtIndexFile

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

- HDB_comparison_library/include/HDB_supergate.hpp
- HDB_comparison_library/src/HDB_supergate.cpp

5.11 HDB_supergate_server_::SERVER Class Reference

Class that contains the DB and is queried upon.

```
#include <HDB_supergate_server.hpp>
```

Public Member Functions

- SERVER (he_cmp::Comparator &comparator, HDB_supergate_::Ctxt_mat &db, HDB_supergate_::CtxtIndexFile &indFile, bool v)
- void **Query** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void **QueryExtensionField** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void **QueryWithIndex** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void testTS (Ctxt &)
- SERVER (he_cmp::Comparator &comparator, HDB_supergate_::Ctxt_mat &db, HDB_supergate_::CtxtIndexFile &indFile, bool v)
- void **Query** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void **QueryExtensionField** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void **QueryWithIndex** (HDB_supergate_::HEQuery &query, HDB_supergate_::Ctxt_mat &result)
- void testTS (Ctxt &)

5.11.1 Detailed Description

Class that contains the DB and is queried upon.

SERVER class contains the encrypted database and the encrypted index file. This simulates the REE.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 SERVER() [1/2]

```
HDB_supergate_server_::SERVER::SERVER (
            he_cmp::Comparator & comparator,
            HDB_supergate_::Ctxt_mat & db,
            HDB_supergate_::CtxtIndexFile & indFile,
            bool v ) [explicit]
```

Constructor of the SERVER class

Parameters

comparator	the reference to comparator class
db	reference to the encrypted database
indFile	reference to the encrypted index file
v	verbose

5.11.2.2 SERVER() [2/2]

```
HDB_supergate_server_::SERVER::SERVER (
             he_cmp::Comparator & comparator,
             HDB_supergate_::Ctxt_mat & db,
             HDB_supergate_::CtxtIndexFile & indFile,
             bool v )  [explicit]
```

Constructor of the SERVER class

Parameters

comparator	the reference to comparator class
db	reference to the encrypted database
indFile	reference to the encrypted index file
v	verbose

5.11.3 Member Function Documentation

5.11.3.1 testTS() [1/2]

```
void HDB_supergate_server_::SERVER::testTS (
             Ctxt &  )
```

debugging function for SERVER::totalSums

5.11.3.2 testTS() [2/2]

```
void HDB_supergate_server_::SERVER::testTS (
             Ctxt & ctxt )
```

debugging function for SERVER::totalSums

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

- HDB_comparison_library/include/HDB_supergate_server.hpp
- HDB_comparison_library/src/HDB_supergate_server.cpp

5.12 HDB_supergate_::STD128_HDB Struct Reference

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.13 HDB_supergate_::TOY_HDB Struct Reference

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

- HDB_comparison_library/include/HDB_supergate.hpp

5.14 HDB_supergate_user_::USER Class Reference

Class that simulates the USER that queries the DB.

```
#include <HDB_supergate_user.hpp>
```

Collaboration diagram for HDB_supergate_user_::USER:

Index