ulpcrypt
API Documentation

General Usage

ulpcrypt compiles to a shared and a static library. To make the function definitions available, just include the header `ulpcrypt.h`. Most functions return the value 0 on success and a negative value, otherwise. Exceptions are the functions for allocating structures, which return a pointer, and the functions for deallocating structures, which return nothing.

Structures

`ulp_public_key`

Public key for U-LP.
Structure members:
`size_t n` - security parameter
`size_t l` - message length
`uint64_t q` - modulus
`uint64_t se` - error bound for encryption
`uint64_t* A` - part of the public key
`uint64_t* P` - part of the public key

`ulp_private_key`

Private key for U-LP.
Structure members:
`size_t n` - security parameter
`size_t l` - message length
`uint64_t q` - modulus
`uint64_t*` S - secret

`ulp_ciphertext`

Ciphertext for U-LP.
Structure members:
`size_t n` - security parameter
`size_t l` - message length
`uint64_t* c1` - first part of the ciphertext
`uint64_t* c2` - second part of the ciphertext

Functions

`ulp_alloc_public_key`

Allocate heap memory for storing a U-LP public key.
Parameters:
`size_t n` - security parameter
`size_t l` - message length
Return value:
`ulp_public_key*` - pointer to the allocated heap memory

ulp_alloc_private_key

Allocate heap memory for storing a U-LP private key.
Parameters:
`size_t n` - security parameter
`size_t l` - message length
Return value:
`ulp_private_key*` - pointer to the allocated heap memory

ulp_alloc_ciphertext

Allocate heap memory for storing a U-LP ciphertext.
Parameters:
`size_t n` - security parameter
`size_t l` - message length
Return value:
`ulp_ciphertext*` - pointer to the allocated heap memory

ulp_free_public_key

Deallocate heap memory for a U-LP public key.
Parameters:
`ulp_public_key* pub_key` - pointer to the memory to free
Return value:
`void`

ulp_free_private_key

Deallocate heap memory for a U-LP private key.
Parameters:
`ulp_private_key* priv_key` - pointer to the memory to free
Return value:
`void`

ulp_free_ciphertext

Deallocate heap memory for a U-LP ciphertext.
Parameters:
`ulp_ciphertext* ciphertext` - pointer to the memory to free
Return value:
`void`

ulp_generate_parameters

Generate the parameters for the U-LP cryptosystem dependent on n and l.
Parameters:
`size_t n` - security parameter
`size_t l` - message length
`uint64_t* sk` - pointer to error bound for key generation (will be generated)
`uint64_t* se` - pointer to error bound for encryption (will be generated)
`uint64_t* q` - pointer to modulus (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

ulp_generate_key_pair

Generate a keypair for the U-LP cryptosystem.
Parameters:
`size_t n` - security parameter
`size_t l` - message length
`uint64_t sk` - error bound for key generation
`uint64_t se` - error bound for encryption

`uint64_t q` - modulus, must be less than 2^{63} due to possible overflow problems
`ulp_public_key** pub_key_p` - pointer to a public key pointer (will be generated)
`ulp_private_key** priv_key_p` - pointer to a private key pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

ulp_encrypt

Encrypt a message with the U-LP cryptosystem.
Parameters:
`uint8_t msg[]` - the bytes to encrypt (number of bits has to match the l parameter in the key)
`ulp_public_key* pub_key` - the public key used for encryption
`ulp_ciphertext** ciphertext_p` - pointer to the ciphertext pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

ulp_decrypt

Decrypt a ciphertext with the U-LP cryptosystem.
Parameters:
`ulp_ciphertext* ciphertext` - pointer to the ciphertext to decrypt
`ulp_private_key* priv_key` - the private key used for decryption
`uint8_t** msg_p` - pointer to the message buffer pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

Ring Structures

ulp_ring_public_key

Public key for U-LP ring variant.
Structure members:
`size_t n` - security parameter
`uint64_t q` - modulus
`uint64_t se` - error bound for encryption
`uint64_t* a` - part of the public key
`uint64_t* p` - part of the public key

ulp_ring_private_key

Private key for U-LP ring variant.
Structure members:
`size_t n` - security parameter
`uint64_t q` - modulus
`uint64_t* s` - secret vector

ulp_ring_ciphertext

Ciphertext for U-LP ring variant.
Structure members:
`size_t n` - security parameter
`uint64_t* c1` - first part of the ciphertext
`uint64_t* c2` - second part of the ciphertext

Ring Functions

ulp_ring_alloc_public_key

Allocate heap memory for storing a U-LP public key (ring variant).
Parameters:
`size_t n` - security parameter and message length

Return value:
`ulp_ring_public_key*` - pointer to the allocated heap memory

ulp_ring_alloc_private_key

Allocate heap memory for storing a U-LP private key (ring variant).
Parameters:
`size_t n` - security parameter and message length
Return value:
`ulp_ring_private_key*` - pointer to the allocated heap memory

ulp_ring_alloc_ciphertext

Allocate heap memory for storing a U-LP ciphertext (ring variant).
Parameters:
`size_t n` - security parameter and message length
Return value:
`ulp_ring_ciphertext*` - pointer to the allocated heap memory

ulp_ring_free_public_key

Deallocate heap memory for a U-LP public key (ring variant).
Parameters:
`ulp_ring_public_key* pub_key` - pointer to the memory to free
Return value:
`void`

ulp_ring_free_private_key

Deallocate heap memory for a U-LP private key (ring variant).
Parameters:
`ulp_ring_private_key* priv_key` - pointer to the memory to free
Return value:
`void`

ulp_ring_free_ciphertext

Deallocate heap memory for a U-LP ciphertext (ring variant).
Parameters:
`ulp_ring_ciphertext* ciphertext` - pointer to the memory to free
Return value:
`void`

ulp_ring_generate_key_pair

Generate a keypair for the U-LP cryptosystem (ring variant).
Parameters:
`size_t n` - security parameter and message length
`uint64_t sk` - error bound for key generation
`uint64_t se` - error bound for encryption
`uint64_t q` - modulus, must be less than 2^{63} due to possible overflow problems
`ulp_ring_public_key** pub_key_p` - pointer to a public key pointer (will be generated)
`ulp_ring_private_key** priv_key_p` - pointer to a private key pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

ulp_ring_encrypt

Encrypt a message with the U-LP cryptosystem (ring variant).
Parameters:
`uint8_t msg[]` - the bytes to encrypt (number of bits has to match the n parameter in the key)
`ulp_ring_public_key* pub_key` - the public key used for encryption

`ulp_ring_ciphertext** ciphertext_p` - pointer to the ciphertext pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise

`ulp_ring_decrypt`

Decrypt a ciphertext with the U-LP cryptosystem (ring variant).
Parameters:
`ulp_ring_ciphertext* ciphertext` - pointer to the ciphertext to decrypt
`ulp_ring_private_key* priv_key` - the private key used for decryption
`uint8_t** msg_p` - pointer to the message buffer pointer (will be generated)
Return value:
`int` - 0 on success, a negative value otherwise