SIDH/Isogeny Signature Function Contracts

Robert Gorrie – McMaster University – December 5, 2017

1 Key Exchange

1.1 Ephemeral Key Generation – Alice

Key generation for Alice	
Location	Efficient Algo's Appendix A
Input	$x_{P_B}, x_{P_A}, y_{P_A}$, $SK_{Alice} = m_A \cdot l_A$
Output	$PK_{Alice} = [x_{\Phi_A}(P_B), x_{\Phi_A}(Q_B), x_{\Phi_A}(Q_B - P_B)]$

EphemeralKeyGeneration_A	
Location	kex.c
Input	unsigned char* PrivateKeyA, unsigned char* PublicKeyA, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	publickey_t PublicKeyA, digit_t PrivateKeyA

1.2 Ephemeral Key Generation – Bob

Key generation for Bob	
Location	Efficient Algo's Appendix A
Input	$x_{P_A}, x_{P_B}, y_{P_B}$, $SK_{Bob} = m_B \cdot l_B$
Output	$PK_{Bob} = [x_{\Phi_B}(P_A), x_{\Phi_B}(Q_A), x_{\Phi_B}(Q_A - P_A)]$

EphemeralKeyGeneration_B	
Location	kex.c
Input	unsigned char* PrivateKeyB, unsigned char* PublicKeyB, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	publickey_t PublicKeyB, digit_t PrivateKeyB

1.3 Ephemeral Secret Agreement – Alice

Shared secret algorithm for Alice	
Location	Efficient Algo's Appendix A
Input	$PK_{Bob} = [x_{\Phi_B}(P_A), x_{\Phi_B}(Q_A), x_{\Phi_B}(Q_A - P_A)]$ $SK_{Alice} = m_A \cdot l_A$
Output	A shared secret j-invariant of an elliptic curve

EphemeralSecretAgreement_A	
Location	kex.c
Input	const unsigned char* PrivateKeyA, const unsigned char* PublicKeyB, unsigned char* SharedSecretA, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	f2elm_t SharedSecretA,

1.4 Ephemeral Secret Agreement – Bob

Shared secret algorithm for Bob	
Location	Efficient Algo's Appendix A
Input	$PK_{Alice} = [x_{\Phi_A}(P_B), x_{\Phi_A}(Q_B), x_{\Phi_A}(Q_B - P_B)]$ $SK_{Bob} = m_B \cdot l_B$
Output	A shared secret j-invariant of an elliptic curve

EphemeralSecretAgreement_B	
Location	kex.c
Input	const unsigned char* PrivateKeyB, const unsigned char* PublicKeyA, unsigned char* SharedSecretB, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	f2elm_t SharedSecretB,

2 Signature Scheme

2.1 Keygen

Keygen		
	Location	Yoo et. al section 4
	Input	security parameter λ
	Output	sk = S, pk = $(E/\langle S\rangle, \Phi(P_B), \Phi(Q_B))$

KeyGeneration_A	
Location	kex.c
Input	unsigned char* PrivateKeyB, unsigned char* PublicKeyB, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	publickey_t PublicKeyB, digit_t PrivateKeyB

KeyGeneration_B	
Location	kex.c
Input	unsigned char* PrivateKeyA, unsigned char* PublicKeyA, PCurveIsogenyStruct CurveIsogeny, invBatch* batch
Output	publickey_t PublicKeyA, digit_t PrivateKeyA

2.2 Sign

Sign	
Location	Yoo et. al section 4
Input	sk = S with order $\ell_A^{e_A}$, message m
Output	$\sigma = ((\texttt{com}_i)_i, (\texttt{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, ((\texttt{resp})[J_i])$

isogny_sign	
Location	SIDH_signature.c
Input	PCurveIsogenyStaticData CurveIsogenyData, unsigned char* PrivateKey, unsigned char* PublicKey, struct Signature* sig
Output	Signature* sig

2.3 Verify

Sign	
Location	Yoo et. al ection 4
Input	pk = $(E/\langle S\rangle, \Phi(P_B), \Phi(Q_B))$, message m, $\sigma = ((\texttt{com}_i)_i, (\texttt{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, ((\texttt{resp})[J_i])$
Output	true or false

isogny_verify	
Location	SIDH_signature.c
Input	PCurveIsogenyStaticData CurveIsogenyData, unsigned char* PublicKey, struct Signature* sig
Output	CRYPTO_STATUS Status

3 Elliptic Curve Operations

4 Field Operations

5 Type Definitions & Structs

alias	definition
digit_t	uint64_t
felm_t	digit_t[NWORDS_FIELD]
f2elm_t	felm_t[2]
publickey_t	f2elm_t[3]

struct	contents	description
signature	unsigned char* Commitments1[NUM_ROUNDS]	
	unsigned char* Commitments2[NUM_ROUNDS]	
	unsigned char* HashResp	
	unsigned char* Randoms[NUM_RUNDS]	
	point_proj* psiS[NUM_ROUNDS]	
PCurveIsogenyStaticData	CurveIsogeny_ID CurveIsogeny	
	unsigned int pwordbits	
	unsigned int owordbits	
	unsigned int pbits	
	digit_t* prime	
	digit_t* A	
	digit_t* C	
	unsigned int oBbits	
	unsigned int eB	
	digit_t* Border	
	digit_t* PA	
	digit_t* PB	
	unsigned int BigMont_A24	
	digit_t* BigMont_order	
	digit_t* Montgomery_R2	
	digit_t* Montgomery_pp	
	digit_t* Montgomery_one	
	RandomBytes RandomBytesFunction	