Constants

P_LENGTH

Q_LENGTH

MP2_HELP_LENGTH = P_LENGTH + Q_LENGTH

CANDIDATE_CODE_LENGTH

SET APDUs	CLA	INS	P1	P2	LC	Data	LE	Observations
SET_P	F0	00	-	-	P_LENGTH	Key modulus p	-	
SET_G	F0	01	-	-	P_LENGTH	Key generator parameter	-	
SET_H	F0	02	-	-	P_LENGTH	Public key	-	
SET_MP_G	F0	03	-	-	P_LENGTH	Exponential message generator	-	
SET_MP_GINV	F0	04	-	-	P_LENGTH	Exponential message generator inverse	-	
SET_Q	F1	00	-	-	Q_LENGTH	Key modulus q	-	
SET_MP2_GV_X	F1	01	-	-	Q_LENGTH	Matrix generator vector Y component	-	
SET_MP2_GV_Y	F1	02	-	-	Q_LENGTH	Matrix generator vector X component	-	
SET_LAMBDA	F1	03	-	-	Q_LENGTH	MP2 Lambda parameter	-	
SET_LAMBDA_MULT	F1	04	-	-	Q_LENGTH	MP2 Lambda multiplier	-	
SET_ALPHA	F2	00	XX	-	-	-	-	P1 = MP alpha parameter

Action APDUs	CLA	INS	P1	P2	LC	Data	LE	Observations
PREPARE_BALLOT	F5	00	XX	XX	-	-	-	P1 = number of candidates; P2 = MarkPledge ballot type (1,2 or 3)
CREATE_CANDIDATE_ENCRYPTION	F5	01	XX	-	-	-	-	P1 = candidate vote index
SELECT_CANDIDATE	F5	02	-	-	CANDIDATE_CODE_LENGTH	candidate code	1	Returns the rotation necessary to align the selected candidate with the YESvote encryption.
PREPARE_RECEIPT	F5	03	-	-	Q_LENGTH	challenge	-	
CREATE_CGS97_CANDIDATE_PROOF	F5	04	XX	-	-	-	-	P1 = candidate vote index
CREATE_MP2_CANONICAL_VOTE	F5	FE	-	-	-	-	-	
CREATE_MP2_CANONICAL_VOTE_WITH_HELP	F5	FF	XX	-	MP2_HELP_LENGTH	g^v(x/y) || -VS(x/y)	-	P1 = selected vector component for the canonical vote (0 => component x; 1 => component y)

Get APDUs	CLA	INS	P1	P2	LC	Data	LE	Observations
GET_PLEDGE	FA	00	-	-	-	-	Q_LENGTH	Returns the pledge value
GET_VCODE	FA	01	XX	-	-	-	Q_LENGTH	Returns the verification code for the candidate index received in P1
GET_VCODE_ENCRYPTION_FACTOR	FA	02	XX	XX	-	-	Q_LENGTH	Returns the encryption factor to verify the correctness of the verification code. P1 selects the candidate index. P2 is used only in MP1 to select the which bit (i.e. BMP) of the vcode is to be verified with the returned value.
GET_CANDIDATE_ENCRYPTION_X	FB	00	XX	XX	-	-	Q_LENGTH	Returns, for the candidate encryption index selected by P1, the encryption X component of the ElGamal encryption specified by P2. Note that the ElGamal encryptions correspont: in a MP1 candidate vote to {canonical vote, BMP0, BMP1, …}, in MP2 P2 selects either the canonical vote or a BMP (i.e. the program automatically selects the revealed element in the BMP); in a MP2 candidate vote to {vector component x, vector component y}; and in MP3 candidate vote to {canonical vote, confirmation code}
GET_CANDIDATE_ENCRYPTION_Y	FB	01	XX	XX	-	-	Q_LENGTH	Returns, for the candidate encryption index selected by P1, the encryption X component of the ElGamal encryption specified by P2. Note that the ElGamal encryptions correspont: in a MP1 candidate vote to {canonical vote, BMP0, BMP1, …}; in a MP2 candidate vote to {vector component x, vector component y}; and in MP3 candidate vote to {canonical vote, confirmation code}
						-		
GET_CGS97_A1	FC	00	-	-	-	-	P_LENGTH	
GET_CGS97_A2	FC	01	-	-	-	-	P_LENGTH	
GET_CGS97_B1	FC	02	-	-	-	-	P_LENGTH	
GET_CGS97_B2	FC	03	-	-	-	-	P_LENGTH	
GET_CGS97_C	FD	00	-	-	-	-	Q_LENGTH	
GET_CGS97_D1	FD	01	-	-	-	-	Q_LENGTH	
GET_CGS97_D2	FD	02	-	-	-	-	Q_LENGTH	
GET_CGS97_R1	FD	03	-	-	-	-	Q_LENGTH	
GET_CGS97_R2	FD	04	-	-	-	-	Q_LENGTH	
GET_SUM_ENCRYPTION_FACTOR	FD	05	-	-	-	-	Q_LENGTH	
GET_MP1_BMP_CONFORMITY_PROOF	FD	FF	XX	XX	-	-	Q_LENGTH	Returns the encryption factor to verify the conformity of the hidden element of the BMP, selected by P2, with the canonical candidate encryption. P1 selects the candidate index.