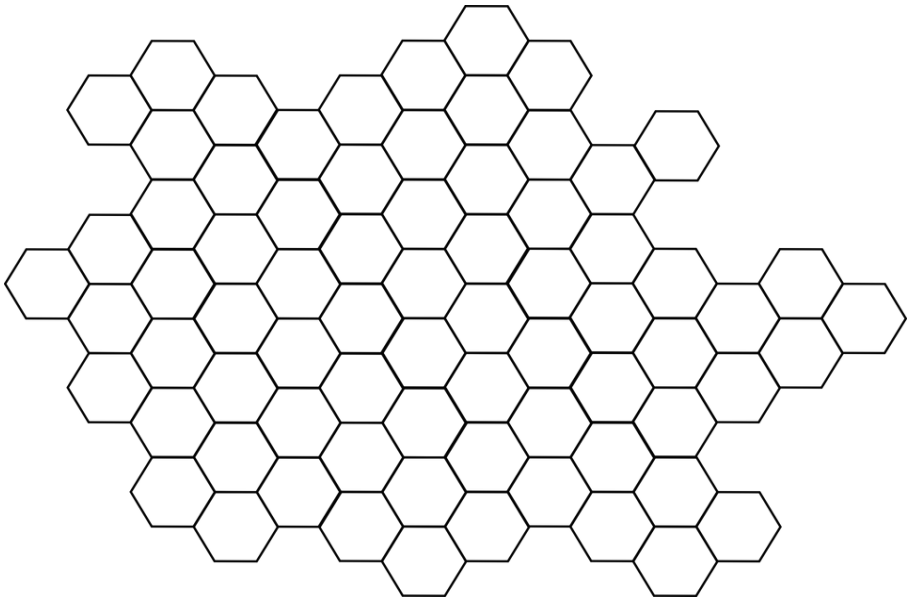
BTOOLS
User Manual
v1.1a

Tools for Blockchains
Transaction Generation for Crypto Currency
Bitcoin and Ethereum
Optionally Secured by a Crypto Currency Smartcard
(CCSC)



Pascal Urien

Content

Sommaire

Introduction

BTOOLS is an open software available for WINDOWS and LINUX, which generates transactions for Bitcoin and Ethereum crypto currency platforms. It is based on the OPENSSL library. In order to provide a strong security, it supports a dedicated *Crypto Currency SmartCard* (CCSC) uses to generate and store secret keys, and to compute cryptographic (ECDSA) signature.

BTOOLS provides the following services:
- Bitcoin address generation (mainnet and testnet);
- Ethereum address generation;
- Bitcoin transaction generation;
- Ethereum transaction generation;
- Simple Bitcoin node client;
- Bitcoin transaction (via the Bitcoin client);
- Ethereum transaction (via WEB APIs);
- Crypto Currency SmartCard scripts for key generation and transaction signature.

Starting BTOOLS

btools

```
usage:
-bin infile outfile, ascii hexa file -> binary file
-dump infile [outfile], binary file -> ascii hexa encoded file
-sha256  infile [outfile], SHA56(file) -> ascii hexa encoded file
-sha3    infile [outfile], SHA3(file)  -> ascii hexa encoded file
-genmain, generate keys and files for bitcoin main network
-gentest, generate keys and files for bitcoin testnet3 network
-editmain privatekey, edit keys and files for bitcoin main network
-edittest privatekey, edit keys and files for bitcoin testnet3 network
-editadr  pubkey [ID], edit bitcoin address from public key and optional ID (hexa)
-checkadr BTC-ADR, check a bitcoin address
-checkwif WIF, check a bitcoin WIF
-gentrans scriptfile transactionfile, generate a transaction file for bitcoin
-checktrans transactionfile, check a transaction file for bitcoin
-mainnet server [timeout(s)], start a client with a bitcoin mainnet server
-testnet server [timeout(s)], start a client with a bitcoin testnet server
-sendmain transaction server [timeout(s)],send a transaction to a mainnet server
-sendtest transaction server [timeout(s)],send a transaction to a testnet server
-decode base58string, decode in hexa a base58 string and verify the checksum
-hexatob58  HexaString, encode an hexa string in a base58 string
-b58tohexa  Base58String, decode a base58 string in an hexa string
-geneth, generate ECDSA keys and files for ethereum network
-editeth privatekey, get ethereum address for ethereum network
-editethadr pubkey, edit ethereum address from public key
-genethtrans scriptfile transactionfile, generate a transaction file for ethereum
-checkethtrans transactionfile, check a transaction file for ethereum
-script filename, run a smartcard script
```

Bitcoin Tools

-genmain

This option generates Bitcoin private and public keys, it computes bitcoin address, hash160, and WIF (*Wallet Import Format*).

Usage:
btools -genmain

Example

btools -genmain

```
PublicKey:
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
PrivateKey 64:
CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081
PublicKey:
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
BTC-Adr: 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
Double SHA2 Check OK
ID: 00
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
BTC-WIF: 5KP4YMxDzfv9P1WVAPZqHRSfi5FydGqqqRjr5oPvskpwTq59wiX
```

-editmain

This option computes Bitcoin public key, bitcoin address, hash160, and WIF (*Wallet Import Format*), from the private key.

Usage:
btools -editmain privatekey

Example

btools -editmain
CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081

```
PublicKey:
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
BTC-Adr: 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
Double SHA2 Check OK
ID: 00
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
BTC-WIF: 5KP4YMxDzfv9P1WVAPZqHRSfi5FydGqqqRjr5oPvskpwTq59wiX
```

-editadr

This option computes Bitcoin address, and hash160 from the public key. It is particularly useful when the public key has been generated from a Crypto Currency Smartcard.

Usage:
btools -editadr pubkey [optional ID]

Example1: mainet (ID=00)

btools -editadr
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E
285BDD3F1FE0A006BD70567885EF57EB149C8880CB9D5AF304182AC942E176CC 00

```
PublicKey:
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9
D5AF304182AC942E176CC
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
BTC-Adr: 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
Double SHA2 Check OK
ID: 00
Hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
```

Example2: testnet (ID=6F)

btools -editadr
0411F1134FAF8C5207CA92D018CF958A26E578281B10BE9FBA0C3AB3B91550B75393
E8B1E8AF974014A063B2E5F552F093F320AD4450750F87FF9B6EC9AC41D99C 6F

```
PublicKey:
0411F1134FAF8C5207CA92D018CF958A26E578281B10BE9FBA0C3AB3B91550B75393E8B1E8AF974014A
063B2E5F552F093F320AD4450750F87FF9B6EC9AC41D99C
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
BTC-Adr: mrq15ZnTdGVbSPzXnGbTRc8nGdr1bTBSP4
Double SHA2 Check OK
ID: 6F
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
```

-checkwif

This option checks a Bitcoin WIF and extracts the associated private key.

Usage:
btools -checkwif WIF

Example1: mainnet WIF (ID=80)

btools -checkwif 5KP4YMxDzfv9P1WVAPZqHRSfi5FydGqqqRjr5oPvskpwTq59wiX

```
Double SHA2 Checksum OK
ID: 80
PrivateKey 64:
CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081
```

Example2: testnet WIF (ID=EF)

btools -checkwif 91oEPVWcUvLjR1gTWFG7Yt6cUq4xRQLViEk6QKAjhKcywLeLNeN

```
Double SHA2 Checksum OK
ID: EF
PrivateKey 64: 1BF601FED4A7AEF29D2F85ED7A18E8BD1E4EC31658F21AB48FD45B4755EA9FF2
```

-gentest

This option generates bitcoin private and public keys, it computes bitcoin address, hash160, and WIF (*Wallet Import Format*) for the testnet network (ID=6F).

Usage:
btools -gentest

Example
btools -gentest

```
PublicKey:
0411F1134FAF8C5207CA92D018CF958A26E578281B10BE9FBA0C3AB3B91550B75393E8B1E8AF974014A
063B2E5F552F093F320AD4450750F87FF9B6EC9AC41D99C
PrivateKey 64:
1BF601FED4A7AEF29D2F85ED7A18E8BD1E4EC31658F21AB48FD45B4755EA9FF2
PublicKey:
0411F1134FAF8C5207CA92D018CF958A26E578281B10BE9FBA0C3AB3B91550B75393E8B1E8AF974014A
063B2E5F552F093F320AD4450750F87FF9B6EC9AC41D99C
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
BTC-Adr: mrq15ZnTdGVbSPzXnGbTRc8nGdr1bTBSP4
Double SHA2 Check OK
ID: 6F
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
BTC-WIF: 91oEPVWcUvLjR1gTWFG7Yt6cUq4xRQLViEk6QKAjhKcywLeLNeN
```

-edittest

This option computes for the testnet, Bitcoin public key, address, hash160, and WIF (Wallet Import Format), from the private key.

Usage:
btools -edittest

Example

btools -edittest
1BF601FED4A7AEF29D2F85ED7A18E8BD1E4EC31658F21AB48FD45B4755EA9FF2

```
PublicKey:
0411F1134FAF8C5207CA92D018CF958A26E578281B10BE9FBA0C3AB3B91550B75393E8B1E8AF974014A
063B2E5F552F093F320AD4450750F87FF9B6EC9AC41D99C
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
BTC-Adr: mrq15ZnTdGVbSPzXnGbTRc8nGdr1bTBSP4
Double SHA2 Check OK
ID: 6F
Hash160: 7C13320AB8BC47F8BD90925C2F7760E9E65C2EC0
BTC-WIF: 91oEPVWcUvLjR1gTWFG7Yt6cUq4xRQLViEk6QKAjhKcywLeLNeN
```

-gentrans

This option generates a bitcoin transaction file from a script. The maximum length of the transaction file is 16384 bytes, with 64 inputs and 64 outputs at the most.

Usage:
btools -gentrans scriptfile transactionfile

The Bitcoin Transaction Script Syntax

Command	Parameter	comment
/		A comment line
*		A comment line
sequence	Four bytes MSB hexadecimal encoding	Default ffffffff
locktime	Four bytes MSB hexadecimal encoding	Default 00000000
nb_input	Integer format	The number of inputs
input		Input separator. It MUST be present for every input.
transaction	32 bytes MSB hexadecimal encoding	The transaction id (UTXO)
index	Integer format	The UTXO index
privkey	32 bytes MSB hexadecimal encoding	The private key
wif	Base58 encoding	The WIF including the private key
apdu_script	A script file name	The script for a smartcard storing the keys and generating the signature
output_nb	Integer format	The number of outputs
output		Output separator. It MUST be present for every output.
output	Ascci string (between quotation marks)	Used for the insertion of OP_RETURN instruction
fee	Decimal format with . separator	Optional fee parameter. Reset for every output. The fee is subtracted from the btc amount. If used it MUST be placed before btc.
btc	Decimal format with . separator	The bitcoin amount of the output.
adr	Base58 encoding	The bitcoin address (payee) of the output.
Hash160	20 bytes MSB hexadecimal encoding	The bitcoin hash160 (payee) of the output.

A script is a set of lines.

A script comment line begins by the '/' or '*' character.

The script defines *sequence* and *locktime* values (hexadecimal MSB encoding)

The number of inputs is specified by the nb_input field

Each input MUST begin by the input field, it comprises:
- a transaction identifier (32 bytes, hexadecimal MSB encoding),
- an index (decimal encoding),
- and a choice between the following fields :
    - privkey [private key hexadecimal MSB encoding],
    - wif [WIF]
    - apdu_script [the name of a smartcard script]

The number of outputs is specified by the nb_output field.

Each output MUST begin by the output field, it comprises
- an optional fee in decimal format, to be subtracted from the btc (i.e. UTXO in most case) value; the character '.' is used as decimal separator
- a btc amount in decimal format, the character '**.**' is used as decimal separator.
- and a choice between the following fields:
    - adr [bitcoin address]
    - hash160 [hash160, hexadecimal MSB encoding]

A special output format includes a string in the OP_RETURN bitcoin instruction:
- output "String". The string is prefixed with its length (OP_RETURN length StringValue)

Example1: one input - one output

```
// bscript01.txt

sequence ffffffff
locktime 00000000

nb_input 1

input
transaction 729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
index       1
privkey     CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081
// wif 5KP4YMxDzfv9P1WVAPZqHRSfi5FydGqqqRjr5oPvskpwTq59wiX
// apdu_script sapdu.txt

nb_output 1

// output "HelloWorld!"
output
// fee 0.0005
btc 0.002685
adr 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
//hash160  CB643DD608FB5C323A4A6342C1A6AC8048B409EB
```

btools -gentrans bscript01.txt btrans01.bin

```
Raw Transaction: Len: 89
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF01D418040000000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000
000001000000
Public Key (input 0):
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Script (input 0): Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
The signature is non canonical...swapping (T,S)
Signature (input 0): Len: 70
304402200772ABD5D37D0CAAB881DBC8912628F93461839CC8D4BC007A355831A6061ED702204CCCC34
B34A9075FC09C9777EAB7A6F5612DA2130C1FF1C0E376AD9B2209D51D
Transaction: Len: 227
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000008
A47304402200772ABD5D37D0CAAB881DBC8912628F93461839CC8D4BC007A355831A6061ED702204CCC
C34B34A9075FC09C9777EAB7A6F5612DA2130C1FF1C0E376AD9B2209D51D014104CFD7A542B8C823992
AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006BD70567885EF57EB149C
8880CB9D5AF304182AC942E176CCFFFFFFFF01D418040000000000001976A914CB643DD608FB5C323A4A6
342C1A6AC8048B409EB88AC0000000001000000

Checking Transaction File btrans01.bin
TransactionId:
D87835CF4AE242657AE59335BEFC63B9B67E3495E4AEC55FFA367053DAC967D2
Version:
00000001
1 inputs
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000001
Input (0) script length checked...
Signature (input 0): Len: 70
304402200772ABD5D37D0CAAB881DBC8912628F93461839CC8D4BC007A355831A6061ED702204CCCC34
B34A9075FC09C9777EAB7A6F5612DA2130C1FF1C0E376AD9B2209D51D
PublicKey (input 0): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182A
C942E176CC
```

```
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Output 0
268500 (satoshi) [D4180400000000000] 0.002685 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Locktime:
00000000
RawTransaction Len: 89
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF01D418040000000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000
000001000000
Signature (0) is OK
The signature (0) is canonical

Transaction file is OK
```

Example2: two inputs - two outputs

```
// bscript02.txt

sequence ffffffff
locktime 00000000

nb_input 2

input
transaction 729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
index       1
privkey     CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081

input
transaction 729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
index       0
wif         5KP4YMxDzfv9P1WVAPZqHRSfi5FydGqqqRjr5oPvskpwTq59wiX

nb_output 2

output
btc 0.000685
adr 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg

output
btc      0.002
hash160  CB643DD608FB5C323A4A6342C1A6AC8048B409EB
```

btools -gentrans bscript02.txt btrans02.bin

```
Raw Transaction: Len: 164
0100000002DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFFDE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E720000000000
FFFFFFFF02940B0100000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC400D0
300000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000
Public Key (input 0):
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Script (input 0: Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
The signature is canonical
Signature (input 0): Len: 70
3044022061FCCD41E740078D99F11195144A637087D3BA176EB320560048AFC76378B23702205D578EB
342F49439317BFC915405CB9B9F61E878FAC43568BF1F4F52367A05F4
Public Key (input 1):
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
```

```
Script (input 1: Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
The signature is non canonical...swapping (T,S)
Signature (input 1): Len: 70
304402201B8BBE48683180A5E660F1E3CDCC51850797A7F9FE6C4520BD11094E881CAEB202206F49F70
CD657F0204788135B31402FD16E125CB1AF9DE8AC25E22FAC48C0A04E
Transaction: Len: 440
0100000002DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000008
A473044022061FCCD41E740078D99F11195144A637087D3BA176EB320560048AFC76378B23702205D57
8EB342F49439317BFC915405CB9B9F61E878FAC43568BF1F4F52367A05F4014104CFD7A542B8C823992
AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006BD70567885EF57EB149C
8880CB9D5AF304182AC942E176CCFFFFFFFFFDE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2
B822CE5014A349E72000000008A47304402201B8BBE48683180A5E660F1E3CDCC51850797A7F9FE6C45
20BD11094E881CAEB202206F49F70CD657F0204788135B31402FD16E125CB1AF9DE8AC25E22FAC48C0A
04E014104CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1F
E0A006BD70567885EF57EB149C8880CB9D5AF304182AC942E176CCFFFFFFFF02940B010000000000197
6A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC400D030000000001976A914CB643DD608
FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000

Checking Transaction File btrans02.bin
TransactionId:
CABD1D75EDFE34FD90521AD0A62791368C61D24E6B828E5E0C28A5044426C910
Version:
00000001
2 inputs
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000001
Input (0) script length checked...
Signature (input 0): Len: 70
3044022061FCCD41E740078D99F11195144A637087D3BA176EB320560048AFC76378B23702205D578EB
342F49439317BFC915405CB9B9F61E878FAC43568BF1F4F52367A05F4
PublicKey (input 0): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000000
Input (1) script length checked...
Signature (input 1): Len: 70
304402201B8BBE48683180A5E660F1E3CDCC51850797A7F9FE6C4520BD11094E881CAEB202206F49F70
CD657F0204788135B31402FD16E125CB1AF9DE8AC25E22FAC48C0A04E
PublicKey (input 1): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Output 0
68500 (satoshi) [940B010000000000] 0.000685 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Output 1
200000 (satoshi) [400D030000000000] 0.002000 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Locktime:
00000000
RawTransaction Len: 164
0100000002DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFFDE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E720000000000
FFFFFFFF02940B010000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC400D0
30000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000
```

```
Signature (0) is OK
The signature (0) is canonical
Signature (1) is OK
The signature (1) is canonical

Transaction file is OK
```

Example3: one input - two outputs including one OP_RETURN instruction

```
// bscript03.txt

sequence ffffffff
locktime 00000000

nb_input 1

input
transaction 729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
index       1
privkey     CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081

nb_output 2

output "Hello World!"

output
fee 0.0005
btc 0.002685
adr 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
```

btools -gentrans bscript03.txt bscript03.bin

```
Raw Transaction: Len: 112
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF020000000000000000E6A0C48656C6C6F20576F726C64218455030000000001976A914CB
643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000
Public Key (input 0):
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB
149C8880CB9D5AF304182AC942E176CC
Script (input 0): Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
The signature is canonical
Signature (input 0): Len: 70
304402206690B8033339DBBFC1FF2A9212827C2D668E20409F3A8BBCCCB2D0C56AE44BDD022068C7977
40884C999A0BE6494F335A7CEDA48A3836794
09D798AFA03997722643
Transaction: Len: 250
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000008
A47304402206690B8033339DBBFC1FF2A9212827C2D668E20409F3A8BBCCCB2D0C56AE44BDD022068C7
97740884C999A0BE6494F335A7CEDA48A383679409D798AFA03997722643014104CFD7A542B8C823992
AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006BD70567885EF57EB149C
8880CB9D5AF304182AC942E176CCFFFFFFFF020000000000000000E6A0C48656C6C6F20576F726C642
18455030000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC00000000010000
00

Checking Transaction File bscript03.bin
TransactionId:
934EFD766EF90EED7FE654A79BA15ABB960307AD67A81A85711F12AE7B0C98B6
Version:
00000001
1 inputs
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000001
```

```
Input (0) script length checked...
Signature (input 0): Len: 70
304402206690B8033339DBBFC1FF2A9212827C2D668E20409F3A8BBCCCB2D0C56AE44BDD022068C7977
40884C999A0BE6494F335A7CEDA48A383679409D798AFA03997722643
PublicKey (input 0): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182A
C942E176CC
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Output 0
0 (satoshi) [0000000000000000] 0.000000 (btc)
Lentgh of script: 14
6A0C48656C6C6F20576F726C6421
Output 1
218500 (satoshi) [8455030000000000] 0.002185 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Locktime:
00000000
RawTransaction Len: 112
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF020000000000000000E6A0C48656C6C6F20576F726C642184550300000000001976A914CB
643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000
Signature (0) is OK
The signature (0) is canonical

Transaction file is OK
```

Example4: Using a Smartcard Script

The script script_key.txt sets public and private keys for the *Crypto Currency SmartCard*. It is executed thanks to the -script option.

```
// script_keys.txt

verbose 1
start
// select AID=010203040500
apdu 00A4040006 010203040500
// Verify Admin PIN
apdu 0020000108 3030303030303030
// Clear KeysPair (index=P2)
apdu 0081 0000 00
// Init Curve (index=P2=0)
apdu 0089 0000 00
// Set PublicKey (index=P2=0)
apdu 0088 0600 41
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3
F1FE0A006BD70567885EF57EB149C8880CB9D5AF304182AC942E176CC
// Set PrivateKey (index=P2=0)
apdu 0088 0700 20
CE1DBAFD7D2E8983ED60E0E081632EB062737B1B1627AAAB276F2E037A74A081
```

btools -script script_keys.txt

```
Opening the APDU script script_keys.txt
Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 81 00 00 00
Rx: 90 00
Tx: 00 89 00 00 00
Rx: 90 00
Tx: 00 88 06 00 41 04 CF D7 A5 42 B8 C8 23 99 2A F5
    1D A8 28 E1 B6 93 CC 5A B6 4F 0C AC F0 F8 0C 31
    A1 EC A4 71 78 6E 28 5B DD 3F 1F E0 A0 06 BD 70
    56 78 85 EF 57 EB 14 9C 88 80 CB 9D 5A F3 04 18
    2A C9 42 E1 76 CC
Rx: 90 00
Tx: 00 88 07 00 20 CE 1D BA FD 7D 2E 89 83 ED 60 E0
    E0 81 63 2E B0 62 73 7B 1B 16 27 AA AB 27 6F 2E
    03 7A 74 A0 81
Rx: 90 00
```

```
// bscript04.txt

sequence ffffffff
locktime 00000000

nb_input 1

input
transaction 729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
index       1
apdu_script sapdu.txt

nb_output 2

output "Hello World!"

output
btc 0.002685
adr 1KYSFr6CyTDMruu8wna981M4ziVyMwftcg
```

The script sapdu.txt is used at runtime. It collects the public key and computes the ECDSA signature from a hash value.

```
// sapdu.txt
verbose 1
start
// select AID= 010203040500
apdu 00A4040006 010203040500
// Verify UserPIN
apdu 0020000004 30303030
// offset of PublicKey in the response, first byte after the 04 prefix
pub 3
// GetKeyParam
apdu 0084 0600 43
// offset of signature in the response
signature 2
// offset of hash to be inserted in the request and thereafter signed
hash 5
// Sign, KeyIndex=P2
apdu 00800000 20
```

btools -gentrans bscript04.txt btrans04.bin

```
Raw Transaction: Len: 112
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF020000000000000000000E6A0C48656C6C6F20576F726C6421D41804000000000001976A914CB
643DD608FB5C323A4A6342C1A6AC8048B409EB88AC00000000001000000

Opening the APDU script sapdu.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 00 43
Rx: 00 41 04 CF D7 A5 42 B8 C8 23 99 2A F5 1D A8 28
    E1 B6 93 CC 5A B6 4F 0C AC F0 F8 0C 31 A1 EC A4
    71 78 6E 28 5B DD 3F 1F E0 A0 06 BD 70 56 78 85
    EF 57 EB 14 9C 88 80 CB 9D 5A F3 04 18 2A C9 42
    E1 76 CC 90 00
Public Key (input 0):
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Script (input 0): Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Opening the APDU script sapdu.txt


Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 00 43
Rx: 00 41 04 CF D7 A5 42 B8 C8 23 99 2A F5 1D A8 28
    E1 B6 93 CC 5A B6 4F 0C AC F0 F8 0C 31 A1 EC A4
    71 78 6E 28 5B DD 3F 1F E0 A0 06 BD 70 56 78 85
    EF 57 EB 14 9C 88 80 CB 9D 5A F3 04 18 2A C9 42
    E1 76 CC 90 00
Tx: 00 80 00 00 20 D5 0D E6 0D 70 8D F6 09 6A 8F A1
    AB A6 33 33 5D 5E C0 46 4E C7 E4 F9 48 CA 76 78
    34 C6 6C A4 9C
Rx: 61 49
Tx: 00 C0 00 00 49
Rx: 00 47 30 45 02 20 05 EC D9 DD 34 71 61 F4 36 6E
    F5 EB C0 B1 D2 06 3E 04 47 1B 79 12 27 C3 A8 CD
    63 A2 81 60 89 77 02 21 00 80 4B BD 34 FA 1B 48
    A5 2E 13 8C 00 7F 20 9E 84 2A 9C FA 85 77 AF 3A
    50 F3 5F DD B0 E6 43 CE FE 90 00
The signature is non canonical...swapping (T,S)
Signature (input 0): Len: 70
3044022005ECD9DD347161F4366EF5EBC0B1D2063E04471B791227C3A8CD63A28160897702207FB442C
B05E4B75AD1EC73FF80DF617A9011E261379965EACC7280DBE9F27243
Transaction: Len: 250
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000008
A4730440220005ECD9DD347161F4366EF5EBC0B1D2063E04471B791227C3A8CD63A28160897702207FB4
42CB05E4B75AD1EC73FF80DF617A9011E261379965EACC7280DBE9F27243014104CFD7A542B8C823992
AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006BD70567885EF57EB149C
8880CB9D5AF304182AC942E176CCFFFFFFFF020000000000000000000E6A0C48656C6C6F20576F726C642
1D41804000000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC000000000010000
00

Checking Transaction File btrans04.bin
TransactionId:
```

```
A3DB0EC12219C3DA90FDCB6EA6643D3F4BEBC05ED6DC6853A8720362B9C45EF0
Version:
00000001
1 inputs
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000001
Input (0) script length checked...
Signature (input 0): Len: 70
3044022005ECD9DD347161F4366EF5EBC0B1D2063E04471B791227C3A8CD63A28160897702207FB442C
B05E4B75AD1EC73FF80DF617A9011E261379965EACC7280DBE9F27243
PublicKey (input 0): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Output 0
0 (satoshi) [0000000000000000] 0.000000 (btc)
Lentgh of script: 14
6A0C48656C6C6F20576F726C6421
Output 1
268500 (satoshi) [D418040000000000] 0.002685 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Locktime:
00000000
RawTransaction Len: 112
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF02000000000000000000E6A0C48656C6C6F20576F726C6421D41804000000000001976A914CB
643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000000001000000
Signature (0) is OK
The signature (0) is canonical

Transaction file is OK
```

-checktrans

This option checks the signature of a bitcoin transaction file.

Usage:
-checktrans transactionfile

Example

btools -checktrans btrans01.bin

```
Checking Transaction File btrans01.bin
TransactionId:
D87835CF4AE242657AE59335BEFC63B9B67E3495E4AEC55FFA367053DAC967D2
Version:
00000001
1 inputs
Input_TxId
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
Input_Index
00000001
Input (0) script length checked...
Signature (input 0): Len: 70
304402200772ABD5D37D0CAAB881DBC8912628F93461839CC8D4BC007A355831A6061ED702204CCCC34
B34A9075FC09C9777EAB7A6F5612DA2130C1FF1C0E376AD9B2209D51D
PublicKey (input 0): Len: 65
04CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E285BDD3F1FE0A006B
D70567885EF57EB149C8880CB9D5AF304182AC942E176CC
```

```
Len: 26
1976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC
Sequence:
FFFFFFFF
Output 0
268500 (satoshi) [D418040000000000] 0.002685 (btc)
To BTC-hash160: CB643DD608FB5C323A4A6342C1A6AC8048B409EB
Locktime:
00000000
RawTransaction Len: 89
0100000001DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72010000000
0FFFFFFFF01D418040000000001976A914CB643DD608FB5C323A4A6342C1A6AC8048B409EB88AC0000
000001000000
Signature (0) is OK
The signature (0) is canonical

Transaction file is OK
```

-mainnet

This option opens a session with a bitcoin node. It is usually used in order to check the availability of a bitcoin P2P server.

Usage:
btools -mainnet server [optional timeout]

Bitcoin useful websites

Bitcoin database: https://blockchain.info/

How to get bitcoin: https://www.coinbase.com

Open source web service API to query the blockchain: https://qbitninja.docs.apiary.io

Bitcoin transaction fees: https://bitcoinfees.21.co/

How to find bitcoin nodes

Bitcoin servers are running on the TCP port 8333.
Here are some methods to get working addresses :

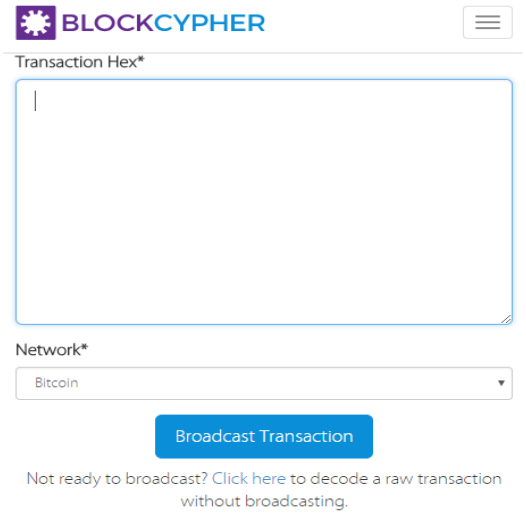
- nslookup bitseed.xf2.org
- online nslookup client: http://www.kloth.net/services/nslookup-fr.php
- https://blockchain.info/ip-log

Example

btools -mainnet 5.178.68.215 30

```
Connecting to 5.178.68.215:8333
sending version message
receiving message: version (length: 104)
sending message verack
receiving message: verack (length: 0)
waiting...
message: alert (length: 168)
message: ping (length: 8)
sending pong message
message: addr (length: 31)
message: inv (length: 1261)
message: inv (length: 1081)
message: inv (length: 289)
message: inv (length: 397)
message: addr (length: 31)
message: inv (length: 793)
message: inv (length: 901)
```

Posting and Checking bitcoin transactions



These web sites post (and verify) bitcoin transactions encoded in hexadecimal format.

- https://blockchain.info/pushtx
- https://blockchain.info/decode-tx

- https://live.blockcypher.com/btc/pushtx
- https://live.blockcypher.com/btc/decodetx/

- https://blockexplorer.com/tx/send

-testnet

This option opens a session with a bitcoin test node. It is usually used in order to check the availability of a bitcoin test P2P server.

Usage:
btools -testnet server [optional timeout(s)]

Testnet useful websites

About testnet: https://en.bitcoin.it/wiki/Testnet

Testnet database: https://www.blocktrail.com/tBTC

How to get bitcoins for testnet: https://testnet.manu.backend.hamburg/faucet

Open source web service API to query the blockchain: https://qbitninja.docs.apiary.io

How to get testnet server address

Testnet servers are running on the TCP port 18333.
Here are some methods to get working addresses :

- https://en.wikipedia.org/wiki/Dig_(command)
- dig online https://www.digwebinterface.com/
- dig commands:
    - dig A *testnet-seed.bitcoin.jonasschnelli.ch*
    - dig A *seed.tbtc.petertodd.org*

Example

btools -testnet 35.187.74.199 30

```
Connecting to 35.187.74.199:18333
sending version message
receiving message: version (length: 102)
sending message verack
receiving message: verack (length: 0)
waiting...
message: alert (length: 168)
message: ping (length: 8)
sending pong message
message: inv (length: 325)
message: addr (length: 31)
message: inv (length: 37)
```

-sendmain

This option sends a transaction file to the bitcoin network. If the operation is OK, no ***reject*** message is received, and the transaction is inserted in the bitcoin transaction pool.
The transaction id is the double SHA256 of the transaction file.

Usage:
btools -sendmain transaction-file server [optional timeout(s)]

Example1: sending a valid transaction

btools -sendmain btrans.bin 5.178.68.215 60

```
Connecting to 5.178.68.215:8333
sending version message
receiving message: version (length: 104)
sending message verack
receiving message: verack (length: 0)
Sending TxId: 06C3E7446E94F380DCAC7FEF9CA7AF8A19C1841BEA4D84232052D2A2D00FD45F
sending transaction message
waiting...
message: ping (length: 8)
sending pong message
message: addr (length: 31)
message: inv (length: 1261)
message: inv (length: 1081)
```

Example2: sending an invalid transaction

btools -sendmain btrans.bin 178.68.215 60

```
Connecting to 5.178.68.215:8333
sending version message
receiving message: version (length: 104)
sending message verack
receiving message: verack (length: 0)
Sending TxId: 06C3E7446E94F380DCAC7FEF9CA7AF8A19C1841BEA4D84232052D2A2D00FD45F
sending transaction message
waiting...
message: ping (length: 8)
sending pong message
message: reject (length: 116)
Type of rejected message: tx
Reason code: REJECT_INVALID
mandatory-script-verify-flag-failed (Script failed an OP_EQUALVERIFY operation)
06C3E7446E94F380DCAC7FEF9CA7AF8A19C1841BEA4D84232052D2A2D00FD45F
```

-sendtest

This option sends a transaction file to the bitcoin test network. If the operation is OK, no ***reject*** message is received, and the transaction is inserted in the bitcoin transaction pool.
The transaction id is the double SHA256 of the transaction file.

Usage:
btools -sendtest transaction-file server [optional timeout(s)]

Example

This transaction has been logged at:
https://www.blocktrail.com/tBTC/tx/7687dd7b036fe75f46bc3e91995086f8f876ce7a6ad0922e14c89710b1017a45

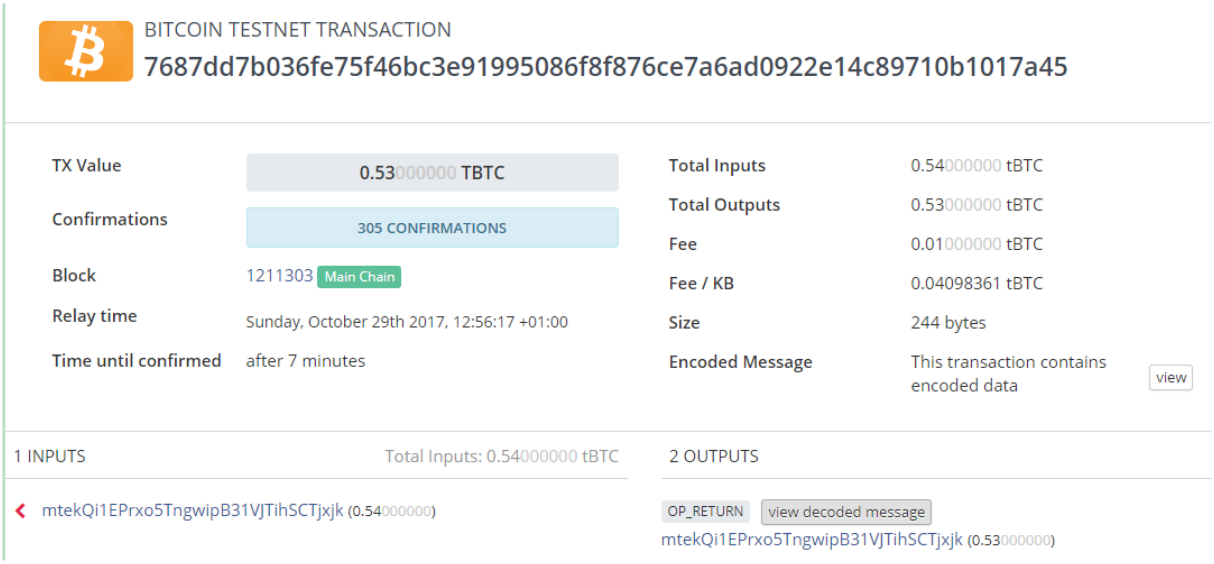
```
// bscript_testnet.txt

sequence ffffffff
locktime 00000000

nb_input 1

input
transaction 9dfea78cf7067a473da571614c3382d361050fea4274bb8d2b18b7bbdc4af44f
index       1
apdu_script sapdu.txt

nb_output 2

output "btools.exe"

output
btc 0.53
adr mtekQi1EPrxo5TngwipB31VJTihSCTjxjk
```

btools -gentrans bscript_testnet.txt testnet.bin

```
Raw Transaction: Len: 110
01000000014FF44ADCBBB7182B8DBB7442EA0F0561D382334C6171A53D477A06F78CA7FE9D010000000
0FFFFFFFF020000000000000000C6A0A62696E77696E2E65786540B728030000000001976A91490130A
A31B920FD68E97365893FC16C8B02449A788AC0000000001000000

Opening the APDU script sapdu.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 00 43
Rx: 00 41 04 B7 CB 52 B0 E4 DA 4C 9F E3 6E FD 72 B5
    10 65 C1 FA AF E1 C7 C9 58 52 6A FB 6D 01 2A 45
    D4 8B 4C ED 75 AC FB 6D 18 75 0C 39 F5 83 93 A2
    21 12 C5 42 FC 00 FC A0 09 D6 D4 21 10 B5 14 CC
    9C CE AB 90 00
Public Key (input 0):
04B7CB52B0E4DA4C9FE36EFD72B51065C1FAAFE1C7C958526AFB6D012A45D48B4CED75ACFB6D18750C3
9F58393A22112C542FC00FCA009D6D42110B514CC9CCEAB
Script (input 0: Len: 26
1976A91490130AA31B920FD68E97365893FC16C8B02449A788AC
Opening the APDU script sapdu.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 00 43
Rx: 00 41 04 B7 CB 52 B0 E4 DA 4C 9F E3 6E FD 72 B5
    10 65 C1 FA AF E1 C7 C9 58 52 6A FB 6D 01 2A 45
    D4 8B 4C ED 75 AC FB 6D 18 75 0C 39 F5 83 93 A2
    21 12 C5 42 FC 00 FC A0 09 D6 D4 21 10 B5 14 CC
    9C CE AB 90 00
Tx: 00 80 00 00 20 8F F8 32 1E 23 AC B9 AF 89 CB 60
    E3 2B 9C 23 A4 30 60 A6 63 97 E6 C1 B3 8C BF 76
    7D 93 29 C4 49
Rx: 61 48
Tx: 00 C0 00 00 48
Rx: 00 46 30 44 02 20 38 B0 12 9F BD 18 78 4C 34 9B
    16 D0 5B 6D D1 58 24 4D 1A DA AA 17 66 03 7D 02
    C2 A7 AB 3C 21 4D 02 20 05 9D 16 FF 4C E9 BA 09
    27 CE 35 75 2D 3E 2E 6D F8 6D BC 79 7F 77 14 A3
    20 F7 02 DF 55 D1 53 5B 90 00
The signature is canonical
Signature (input 0): Len: 70
3044022038B0129FBD18784C349B16D05B6DD158244D1ADAAA1766037D02C2A7AB3C214D0220059D16F
F4CE9BA0927CE35752D3E2E6DF86DBC797F7714A320F702DF55D1535B
Transaction: Len: 248
01000000014FF44ADCBBB7182B8DBB7442EA0F0561D382334C6171A53D477A06F78CA7FE9D010000008
A473044022038B0129FBD18784C349B16D05B6DD158244D1ADAAA1766037D02C2A7AB3C214D0220059D
16FF4CE9BA0927CE35752D3E2E6DF86DBC797F7714A320F702DF55D1535B014104B7CB52B0E4DA4C9FE
36EFD72B51065C1FAAFE1C7C958526AFB6D012A45D48B4CED75ACFB6D18750C39F58393A22112C542FC
00FCA009D6D42110B514CC9CCEABFFFFFFFF020000000000000000C6A0A62696E77696E2E65786540B
728030000000001976A91490130AA31B920FD68E97365893FC16C8B02449A788AC0000000001000000

Checking Transaction File testnet.bin
TransactionId:
7687DD7B036FE75F46BC3E91995086F8F876CE7A6AD0922E14C89710B1017A45
...
Transaction file is OK
```

btools -sendtest testnet.bin 172.104.59.47 60

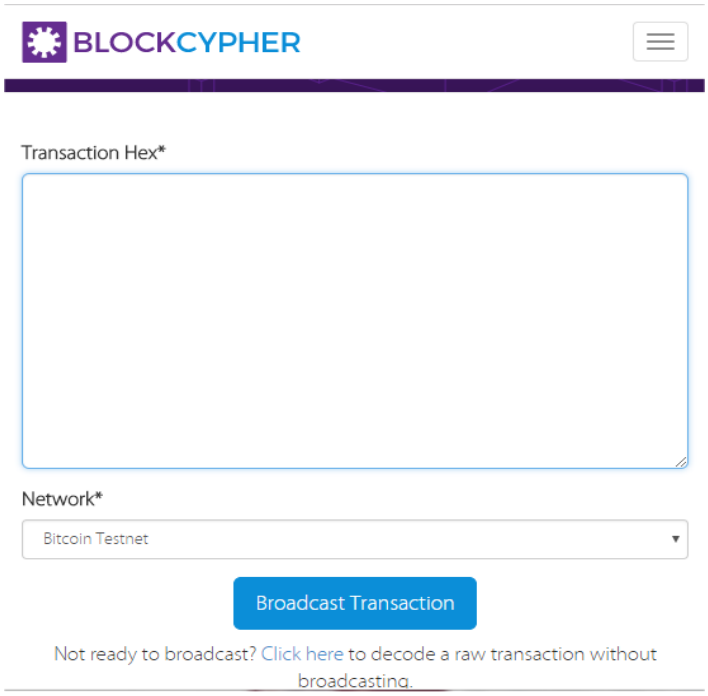
```
Connecting to 172.104.59.47:18333
sending version message
receiving message: version (length: 112)
sending message verack
receiving message: verack (length: 0)
Sending TxId: 7687DD7B036FE75F46BC3E91995086F8F876CE7A6AD0922E14C89710B1017A45
sending transaction message
waiting...
```



Posting and Checking testnet transactions

These web sites post (and verify) testnet transactions encoded in hexadecimal format.

- https://live.blockcypher.com/btc-testnet/pushtx/
- https://live.blockcypher.com/btc-testnet/decodetx/



Ethereum Tools

-geneth

This option generates Ethereum public and private keys, and the Ethereum associated address.

Usage:
btools -geneth

Example

btools -geneth

```
PublicKey:
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
PrivateKey 64: E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
Ether Address: 777A07BAB1C119D74545B82A8BE72BEAFF4D447B
```

-editheth

This option edits the public key and the ethereum address from the private key

Usage:
btools -editeth privatekey

Example

btools -editeth
E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042

```
Pub:
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
Ether Address: 777A07BAB1C119D74545B82A8BE72BEAFF4D447B
```

-editethadr

This option edits the ether address from the public key.

Usage:
btools -editethadr pubkey

Example

btools -editethadr
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF5
2581076023A8B33D8BA6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6

```
Public Key:
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8BA6F8E7E89EC
1C5F0D66B1EFFC744582AF063187297592F6
Ether Address: 777A07BAB1C119D74545B82A8BE72BEAFF4D447B
```

-genethtrans

This option generates an Ethereum transaction file from a script file. The maximum length of the transaction file is 16384 bytes.

Usage:
btools -genethtrans scriptfile transactionfile

Ethereum Transaction File Syntax

A script is a set of lines.

A script comment line begins by the '/' or '*' character.

The script file comprises the following elements:

- the private key (in hexadecimal format) or the name of a smartcard script. The bitcoin and Ethereum smartcard script follow the same syntax.

- a *nonce* field. the nonce is expressed in decimal format.

- a *gasPrice* field. The *gasPrice*, in WEI unit (1 WEI = 10^{-18} ether) is an integer.

- a *gasLimit* field. The *gasLimit*, in WEI unit (1 WEI = 10^{-18} ether) is an integer.

- the *to* field indicates the ether destination address. It is 20 bytes hexadecimal value.

- a *value* field indicates the transaction amount. The *amount*, in WEI unit (1 WEI = 10^{-18} ether) is a decimal integer. The maximum value is about 16 ethers.

- a *data* field. Three options are available:
    - data, text (ascii) data field
    - datab, hexadecimal data field
    - dataf, a binary file

Command	Parameter	Comment
privkey	20 bytes MSB hexadecimal encoding	The private key
apdu_script	The script file name	The script for a smartcard storing the keys and generating the signature
nonce	Integer format	The nonce of the transaction. A zero vale is encoded as null (0x80)
gasPrice	Integer format	The gas price of the transaction in WEI
gasLimit	Integer format	The gas limit of the transaction in WEI
to	20 bytes MSB hexadecimal encoding	The Ethereum address (payee) of the transaction
value	Integer format	The amount of the transaction in WEI. the maximum is about 16 Ethers
data	Ascii string (between quotation marks)	The data of the transaction expressed in ascii text. Empty parameter in encoded as a null (0x80) value
datab	Hexadecimal encoding	The data of the transaction expressed in hexadecimal format. Empty parameter in encoded as a null (0x80) value
dataf	File name	The data of the transaction loaded from a file format. Empty parameter in encoded as a null (0x80) value

Example1: Empty Data Field

```
// escript01.txt

privkey   E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
// apdu_script sapdu.txt

nonce     0
gasPrice  21000000000
gasLimit  40000
to        777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value     10000000000000000
data
// data "HelloWorld"
// datab F10203A4
// dataf file.bin
```

btools -genethtrans escript01.txt escript01.bin

```
Opening the script escript01.txt

nonce: 0
gazPrice: 04E3B29200
gazLimit: 9C40
to: 777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value: 10000000000000000 Wei (0.010000 ETH)
Raw Transaction Len: 41
E8808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC1000080

PublicKey:
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
PrivateKey 64:
E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
The signature is canonical
R: 02F1DD7D3B245D75368B467B06CAD6100267031935B7474ACB5C74FE7D8C904097
Q:
04D896BE505CADA79F704FD94912C4E2BBAD52F7CAE6A5F07311D30CBC889574D2A06FA6F9E429CC761
B0AE10429C9575197EA47A471F4491D41D903349B30F3F5
Signature Verified
R: 03F1DD7D3B245D75368B467B06CAD6100267031935B7474ACB5C74FE7D8C904097
Q:
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
Signature Verified
PublicKey Recovered, v=28

Final Transaction Len: 109
F86B808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC100008
01CA0F1DD7D3B245D75368B467B06CAD6100267031935B7474ACB5C74FE7D8C904097A0772D65407480
D7C45C7E22F84211CB1ADF9B3F36046A2F93149135CADBB9385D

Checking Transaction File escript01.bin
TransactionId: Len: 32
F8F544E06D40C101CEF04BDEDEC38089FF1FBF218B7034625603661605E36EBB
List Length: 107
nonce: 0x
gasPrice: 21000000000 (21 GWei)
gasLimit: 40000
to: 0x777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value:: 0.010000 ETH (10000000000000000 Wei)
data: 0x
v: 28
r: 0xF1DD7D3B245D75368B467B06CAD6100267031935B7474ACB5C74FE7D8C904097(32)
```

```
s: 0x772D65407480D7C45C7E22F84211CB1ADF9B3F36046A2F93149135CADBB9385D(32)
Raw Transaction: Len: 41
E8808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC1000080
Signature Verified

From: PublicKey=
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
From: ETH Address= Len: 20
777A07BAB1C119D74545B82A8BE72BEAFF4D447B
the file escript01.bin has been successfully verified
```

Other Script Examples

btools -genethtrans escript02.txt escript02.bin
btools -genethtrans escript03.txt escript03.bin
btools -genethtrans escript04.txt escript04.bin

```
// escript02.txt

privkey   E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
// apdu_script sapdu.txt

nonce     0
gasPrice  21000000000
gasLimit  40000
to        777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value     10000000000000000
data "HelloWorld"
```

```
// escript03.txt

privkey   E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
// apdu_script sapdu.txt

nonce     0
gasPrice  21000000000
gasLimit  40000
to        777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value     10000000000000000
datab     F10203A4
```

```
// escript04.txt

privkey   E49344BD32802138C9A250FCEA13F6AE30E17BC945F107F05618AFC0ED523042
// apdu_script sapdu.txt

nonce     0
gasPrice  21000000000
gasLimit  40000
to        777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value     10000000000000000
dataf     file.bin
```

-checkethtrans

This option checks an Ethereum transaction file and verifies its signature

Syntax:
btools -checkethtrans transactionfile

Example

btools -checkethtrans escript01.bin

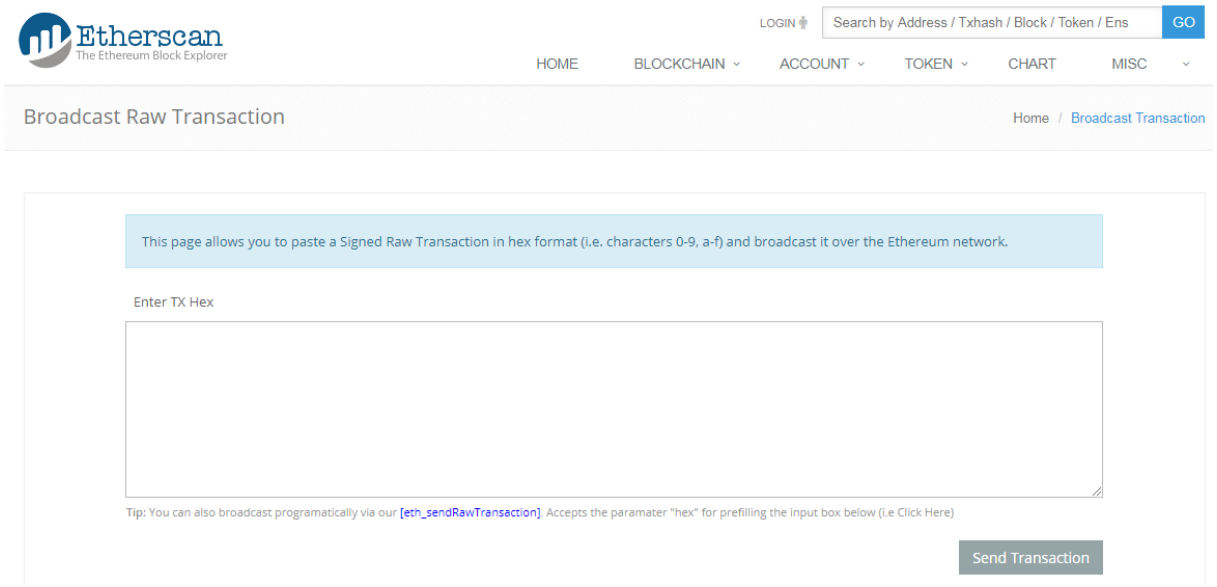
```
Checking Transaction File escript01.bin
TransactionId: Len: 32
F8F544E06D40C101CEF04BDEDEC38089FF1FBF218B7034625603661605E36EBB
List Length: 107
nonce: 0x
gasPrice: 21000000000 (21 GWei)
gasLimit: 40000
to: 0x777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value:: 0.010000 ETH (10000000000000000 Wei)
data: 0x
v: 28
r: 0xF1DD7D3B245D75368B467B06CAD6100267031935B7474ACB5C74FE7D8C904097(32)
s: 0x772D65407480D7C45C7E22F84211CB1ADF9B3F36046A2F93149135CADBB9385D(32)
Raw Transaction: Len: 41
E8808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC1000080
Signature Verified

From: PublicKey=
0477AAA9AE8ADCAAA26F930D6022E470BBC16E10AF22A5482DAB0798A5A2C2AF52581076023A8B33D8B
A6F8E7E89EC1C5F0D66B1EFFC744582AF063187297592F6
From: ETH Address= Len: 20
777A07BAB1C119D74545B82A8BE72BEAFF4D447B
the file escript01.bin has been successfully verified
```

Sending Ethereum Transactions

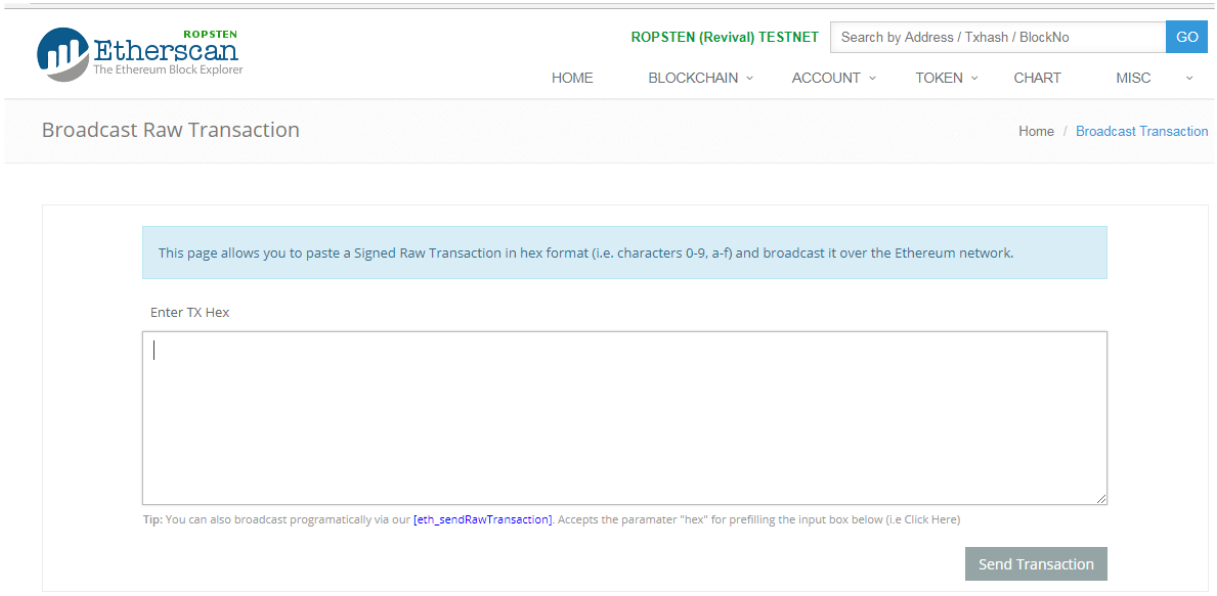
Transaction files expressed in hexadecimal can be *cut and paste* in WEB API performing transaction operation with Ethereum network or test (rosten).

Website for Sending Ethereum Transactions

For ethereum:
https://etherscan.io/pushTx



For the ropsten test network:
https://ropsten.etherscan.io/pushTx



How to get Ethers for the ROSTEN test network
- Install the *metamask* plugin, https://metamask.io/

Example

```
// escript05.txt

apdu_script sapdu1.txt

nonce     10
gasPrice  21000000000
gasLimit  40000
to        3f406a15095669e63df80d21d54d12bdfa214187
value     10000000000000000
data      "btools.exe"
```

btools -genethtrans escript05.txt etrans05.bin

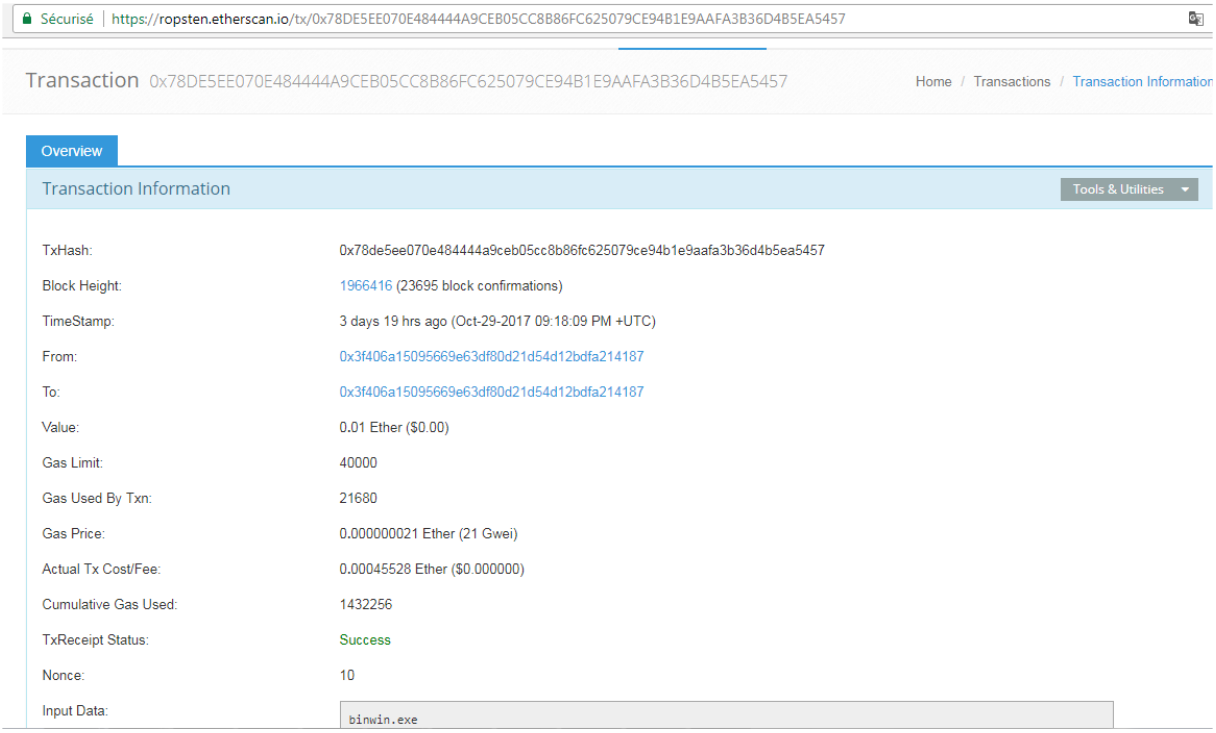
```
Opening the script escript05.txt

nonce: 0A
gazPrice: 04E3B29200
gazLimit: 9C40
to: 3F406A15095669E63DF80D21D54D12BDFA214187
value: 10000000000000000 Wei (0.010000 ETH)
data: btools.exe
Raw Transaction Len: 51
F20A8504E3B29200829C40943F406A15095669E63DF80D21D54D12BDFA214187872386F26FC100008A6
2696E77696E2E657865

Opening the APDU script sapdu1.txt


Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 01 43
Rx: 00 41 04 C9 60 0E 08 42 54 75 75 25 DA D8 FC FF
    8C A9 99 AA 67 E6 B9 E0 AC 98 8A 86 2F B0 79 5E
    38 5F 42 B8 4E EF 83 96 54 4C 0B 3B 3A 1A 61 71
    E3 00 81 65 77 7B F4 F2 3E 8E 98 54 1C E2 2C 5B
    6B DF 3A 90 00
Tx: 00 80 00 01 20 6E F6 03 7E 62 BA D3 87 0D 0A 81
    28 82 BC DC 9C 52 FF 60 23 7C A5 6D 69 EC 23 1F
    A6 39 E1 06 C1
Rx: 61 49
Tx: 00 C0 00 00 49
Rx: 00 47 30 45 02 20 6A 83 C2 E7 C0 4B 46 88 91 F6
    57 E9 A4 6D DA D8 0B E2 0D 74 94 47 AC 37 2E F0
    7E 51 DD F4 8C 81 02 21 00 B2 91 18 32 0B 37 37
    D5 61 49 27 DC 83 62 09 41 01 F5 6F 78 60 97 AE
    0C 7A 46 92 1A 19 9B E9 C3 90 00
PublicKey:
04C9600E084254757525DAD8FCFF8CA999AA67E6B9E0AC988A862FB0795E385F42B84EEF8396544C0B3
B3A1A6171E3008165777BF4FE8E98541CE22C5B6BDF3A
The signature is non canonical...swapping (T,S)
R: 026A83C2E7C04B468891F657E9A46DDAD80BE20D749447AC372EF07E51DDF48C81
Q:
04C9600E084254757525DAD8FCFF8CA999AA67E6B9E0AC988A862FB0795E385F42B84EEF8396544C0B3
B3A1A6171E3008165777BF4F23E8E985CE22C5B6BDF3A
Signature Verified
PublicKey Recovered, v=27


Final Transaction Len: 119
```

```
F8750A8504E3B29200829C40943F406A15095669E63DF80D21D54D12BDFA214187872386F26FC100008
A62696E77696E2E6578651BA06A83C2E7C0468891F657E9A46DDAD80BE20D749447AC372EF07E51DDF4
8C81A04D6EE7CDF4C8C82A9EB6D8237C9DF6BDB8B96D6E4EB0F22F458BCC72B69A577E
```

Checking Transaction File etrans05.bin

```
TransactionId: Len: 32
78DE5EE070E484444A9CEB05CC8B86FC625079CE94B1E9AAFA3B36D4B5EA5457
...
From: PublicKey=
04C9600E084254757525DAD8FCFF8CA999AA67E6B9E0AC988A862FB0795E385F42B84EEF8396544C0B3
B3A1A6171E30081657BF4F23E8E98541CE22C5B6BDF3A
From: ETH Address= Len: 20
3F406A15095669E63DF80D21D54D12BDFA214187
the file etrans05.bin has been successfully verified
```

https://ropsten.etherscan.io/tx/0x78DE5EE070E484444A9CEB05CC8B86FC625079CE94B1E9AAFA3B36D4B5EA5457



Useful Ethereum websites

Ethereum blockchain: https://etherscan.io/

Ethereum GAS station: https://ethgasstation.info/

Smartcard tools

Regarding the smartcard use, the btools software is based on the PCSC (for Windows) and PCSC-Lite (for Linux) libraries.

-script

This option starts a script for smartcard use.

Usage:
btools -script scriptname

Script Commands

Command	Parameter	Comment
verbose	0 or 1	Set/Cancel the verbose mode (default no=0)
test	0 or 1	Set/Cancel the test mode (default no=0) In the normal mode of execution the script is stopped at the first error, i.e. SW1,SW2 different from 90,00
start	AID hexadecimal encodind (optional)	Start the first available smartcard
apdu	adpu in hexadecimal format	Send an ISO7816 request to the smartcard. In normal operation the response should end by the 9000 status
pub	offset, integer	MUST be specified before the apdu used to collect the public key. It is the offset in the response of the public key (after the byte 04) (default=3)
signature	offset, integer	MUST be specified before the apdu used to collect the signature. It is the offset in the response of the ASN.1 encoding of the ECDSA signature (default 2)
hash	offset, integer	MUST be specified before the apdu used to collect the signature. It is the offset in the ISO7616 request of the hash (or data) to be signed (default 5)
scsize	offset, integer	MUST be specified before the apdu used to collect the card status (default 4). It is the offset in the response of the Read/Write memory size (2 bytes MSB-LSB) i.e. maxsize
maxsize	size, integer	Use to manually set the Read/Write memory size. Not needed when scsize is available (default value 3*1024)
scwrite	filename	Write a file in the Read/Write memory. the first two bytes written indicate the filesize
scread	filename	Read the Read/Write memory in a file.
scdisplay		Display in text format the content of the Read/Write memory

A script is a set of lines.

A script comment line begins by the '/' or '*' character.

A simple Script

```
// scheck.txt
verbose 1
start
// select
apdu 00A4040006 010203040500
// UserPin= 0000
apdu 0020 0000 04 30303030
// Get Status
apdu 0087 0000 06
```

btools -script scheck.txt

```
Opening the APDU script scheck.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 87 00 00 06
Rx: 01 08 00 03 0C 00 90 00
```

Keys Generation Script

```
// sgen5.txt
verbose 1
start
// select
apdu 00A4040006 010203040500
// Verify PinAdmin
apdu 0020 0001 08 30 30 30 30 30 30 30 30
// Curve 0 SECP256k1, Keys Generation index=5
// ClearKeys Key5
apdu 0081 00 05 00
// InitCurve 0, Key5
apdu 0089 00 05 00
// GenerateKeysPair Key5
apdu 0082 00 05 00
// Dump KeysPair Key5
apdu 0083 00 05 02
// GetPublicKey Key5
apdu 0084 06 05 00
// GetPrivateKey Key5
apdu 0084 07 05 00
```

btools -script sgen5.txt

```
Opening the APDU script sgen5.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
```

```
Rx: 90 00
Tx: 00 81 00 05 00
Rx: 90 00
Tx: 00 89 00 05 00
Rx: 90 00
Tx: 00 82 00 05 00
Rx: 90 00
Tx: 00 83 00 05 02
Rx: 01 85 90 00
Tx: 00 84 06 05 00
Rx: 6C 43
Tx: 00 84 06 05 43
Rx: 00 41 04 D1 66 79 10 8B 71 21 1B 31 00 20 27 11
    DF 21 9B 87 C3 49 62 6F 6D DB 4E A6 71 79 B0 7C
    AD 48 A1 9F 5A 43 13 24 0A 50 96 AD BB 59 86 D5
    1B 9F 19 67 84 19 9E 6B 3E 35 0C 40 EE A5 1F 72
    87 7B CD 90 00
Tx: 00 84 07 05 00
Rx: 6C 22
Tx: 00 84 07 05 22
Rx: 00 20 0E 83 16 35 62 BF E0 2E 3F 37 16 A2 38 4B
    E5 0F 6D 3A 88 BA C1 A8 63 96 84 06 6F EE 25 31
    50 8A 90 00
```

Keys Setting Script

```
// sstore5.txt
verbose 1
start
// select
apdu 00A4040006 010203040500
// Verify PinAdmin
apdu 0020 0001 08 30 30 30 30 30 30 30 30
// ClearKeys Key5
apdu 0081 00 05 00
// InitCurve 0, Key5
apdu 0089 00 05 00
// SetPublicKey Key5
apdu 0088 06 05 41
04A6FC0C5F467C3DB8C1581805E7C62C5FAEA19063B01F5845AD68DE9D84385F321EBF3A26B299
12418992DCDC1FE69C282EFF65860E109F53AD27A29624984B6A
// SetPrivateKey Key5
apdu 0088 07 05 20
17439172198780D8635D1B2259FF35AC8AACCAE56D914D20C9B9D19D20D029DB
```

btools -script sstore5.txt

```
Opening the APDU script sstore5.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 81 00 05 00
Rx: 90 00
Tx: 00 89 00 05 00
Rx: 90 00
Tx: 00 88 06 05 41 04 A6 FC 0C 5F 46 7C 3D B8 C1 58
    18 05 E7 C6 2C 5F AE A1 90 63 B0 1F 58 45 AD 68
    DE 9D 84 38 5F 32 1E BF 3A 26 B2 99 12 41 89 92
    DC DC 1F E6 9C 28 2E FF 65 86 0E 10 9F 53 AD 27
    A2 96 24 98 4B 6A
```

```
Rx: 90 00
Tx: 00 88 07 05 20 17 43 91 72 19 87 80 D8 63 5D 1B
    22 59 FF 35 AC 8A AC CA E5 6D 91 4D 20 C9 B9 D1
    9D 20 D0 29 DB
Rx: 90 00
```

Transaction Script

```
// ssign5.txt

verbose 1
start
// Select CCSC

apdu 00A4040006 010203040500
// Verify UserPIN
apdu 0020000004 30303030

// Get PublicKey 5
pub 3
apdu 0084 0605 43

// ECDSA Sign
signature 2
hash 5
apdu 0080 0005 20
```

```
// escript01_5.txt

apdu_script ssign5.txt
nonce       0
gasPrice    21000000000
gasLimit    40000
to          777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value       10000000000000000
data
```

btools -genethtrans escript01_5.txt ecsript_01_5.bin

```
Opening the script escript01_5.txt

nonce: 0
gazPrice: 04E3B29200
gazLimit: 9C40
to: 777A07BAB1C119D74545B82A8BE72BEAFF4D447B
value: 10000000000000000 Wei (0.010000 ETH)
data: NULL
Raw Transaction Len: 41
E8808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC1000080

Opening the APDU script ssign5.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 84 06 05 43
Rx: 00 41 04 A6 FC 0C 5F 46 7C 3D B8 C1 58 18 05 E7
    C6 2C 5F AE A1 90 63 B0 1F 58 45 AD 68 DE 9D 84
    38 5F 32 1E BF 3A 26 B2 99 12 41 89 92 DC DC 1F
```

```
    E6 9C 28 2E FF 65 86 0E 10 9F 53 AD 27 A2 96 24
    98 4B 6A 90 00
Tx: 00 80 00 05 20 DC AF B4 6D 7F 57 1D 87 C2 34 B3
    20 8E 68 86 AD F4 85 AC 98 20 EA A5 67 7C 6D 37
    6A 32 13 6F 34
Rx: 61 48
Tx: 00 C0 00 00 48
Rx: 00 46 30 44 02 20 65 A3 1E 14 88 20 61 82 1E A8
    B7 27 C4 A8 D1 E2 CB 59 29 20 88 6B DD 70 84 B9
    C1 C5 D6 6F 7D 30 02 20 5B 83 A4 69 E5 6D 3B B1
    C2 77 6B 16 A3 7B C1 19 0F 6A C9 85 F7 03 54 B6
    58 1B 6F 46 21 C7 63 3B 90 00
PublicKey:
04A6FC0C5F467C3DB8C1581805E7C62C5FAEA19063B01F5845AD68DE9D84385F321EBF3A26B29912418
992DCDC1FE69C282EFF65860E109F53AD27A29624984B6A
...
Final Transaction Len: 109
F86B808504E3B29200829C4094777A07BAB1C119D74545B82A8BE72BEAFF4D447B872386F26FC100008
01CA065A31E14882061821EA8B727C4A8D1E2CB592920886BDD7084B9C1C5D66F7D30A05B83A469E56D
3BB1C2776B16A37BC1190F6AC985F70354B6581B6F4621C7633B
```

Read/Write Script

```
// srw.txt

verbose 1
start
// Select CCSC
apdu 00A4040006 010203040500
// Verify User2 PIN
apdu 0020000204 30303030

// maxsize 3072

scsize 4
apdu 00870000 06

scwrite   hello.txt
scread    hello2.txt
scdisplay
```

btools -script srw.txt

```
Opening the APDU script srw.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 02 04 30 30 30 30
Rx: 90 00
Tx: 00 87 00 00 06
Rx: 01 08 00 20 0C 00 90 00
Tx: 00 D0 00 00 0D 00 0B 48 65 6C 6C 6F 57 6F 72 6C
    64 21
Rx: 90 00
Tx: 00 B0 00 00 02
Rx: 00 0B 90 00
Tx: 00 B0 00 02 0B
Rx: 48 65 6C 6C 6F 57 6F 72 6C 64 21 90 00
Tx: 00 B0 00 00 02
Rx: 00 0B 90 00
Tx: 00 B0 00 02 0B
Rx: 48 65 6C 6C 6F 57 6F 72 6C 64 21 90 00
BEGIN
HelloWorld!
END
```

Software Tools

These software tools are useful in order to perform operations in the crypto currency context, and to exchange cryptographic information with OPENSSL.

-bin

This option converts a text file in a binary file

Usage:
btools -bin infile outfile

Example

btools -bin sign.txt sign.bin

```
71 bytes have been written
```

```
30 45
02 21
00 E4 1B AA AE 68 20 E7 60 45 EB E3 08 ED 77 60 A9
8E 83 5A D1 7C E2 58 AE FA 6B 15 7E 8C B2 A0 A6
02 20
4D F3 A0 EC 99 8C 61 77 BA A6 3E 67 53 06 AB 47 A2
08 03 BE E2 AA 34 DA A3 25 9C D4 C2 0D DC C2

The sign.txt file (ECDSA signature, ASN1 encoding)
```

-dump

This option dumps a binary file in a hexadecimal text format.

Usage:
btools -dump infile [outfile]

Example1

btools -dump sign.bin

```
71 bytes
3045022100E41BAAAE6820E76045EBE308ED7760A98E835AD17CE258AEFA6B157E8CB2A0A602204DF3A
0EC998C6177BAA63E675306AB47A20803BEE2AA34DAA3259CD4C20DDCC2
```

Example2

btools -dump sign.bin signa.txt

```
signa.txt
3045022100E41BAAAE6820E76045EBE308ED7760A98E835AD17CE258AEFA6B157E8CB2A0A602204DF3A
0EC998C6177BAA63E675306AB47A20803BEE2AA34DAA3259CD4C20DDCC2
```

-sha256

This option computes the SHA256 digest.

Usage:
btools -sha256 infile [outfile]

Example1

btools -sha256 hello.txt
```
file hello.txt has 11 bytes
SHA256:
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
```

Example2

btools -sha256 hello.txt digest.bin
```
file hello.txt has 11 bytes
SHA256 written to file digest.bin
```

btools -dump hello.txt
```
11 bytes
48656C6C6F576F726C6421   //HelloWorld!
```

btools -dump digest.bin
```
32 bytes
729E344A01E52C822BDFDEC61E28D6EDA02658D2E7D2B80A9B9029F41E212DDE
```

-sha3

This option computes the sha3 (*keccak*) digest

Usage:
btools -sha3 infile [outfile]

Example1

btools -sha3 hello.txt
```
file hello.txt has 11 bytes
SHA3:
2A901ED9A0877F6D161C62E87D7D73A0942C8078640507854C5D66EFC2819897
```

Example2

btools -sha3 hello.txt digest.bin
```
file hello.txt has 11 bytes
SHA3 written to file digest.bin
```

Example3

btools -dump digest.bin
```
32 bytes
2A901ED9A0877F6D161C62E87D7D73A0942C8078640507854C5D66EFC2819897
```

-hexatob58

This option performs hexadecimal (base16) conversion to base58

Syntax:
btools -hexatob58 HexaString

Example

btools -hexatob58
2A901ED9A0877F6D161C62E87D7D73A0942C8078640507854C5D66EFC2819897
```
3s9atJAwp15YdRLH2rKXBMVS3ya73BDS8ijheFNErEd4
```

-b58tohexa

This option performs conversion to base58 hexadecimal (base16).

Syntaxe:
btools -b58tohexa   Base58String

Example

btools -b58tohexa 3s9atJAwp15YdRLH2rKXBMVS3ya73BDS8ijheFNErEd4
2A901ED9A0877F6D161C62E87D7D73A0942C8078640507854C5D66EFC2819897

The Crypto Currency SmartCard

The Crypto Currency smartcard application (CCSC), of which AID is 010203040500 has three PINs, administrator, user, and user2. The default values are 8 zeros (3030303030303030) for administrator and 4 zeros (30303030) for user and user2.

It is able to generate or to import elliptic curve keys (up to 8), used for the generation of ECDSA signatures used by Bitcoin and Ethereum crypto currencies.

A Read/Write non volatile memory, protected by a dedicated PIN (User2), is available for the storage of any sensitive information.

The Select Command

This commands starts the Crypto Currency smartcard application
Upon success it returns the status word SW1 SW2 = 9000

Command

CLA	INS	P1	P2	P3	AID
00	A4	04	00	06	010203040500

Response

SW1	SW2
90	00

The Verify UserPin Command

This command verifies the user pin. The UserPin is required for the signature operations.

Upon success it returns the status word SW1 SW2 = 9000
Otherwise it returns SW1=63, SW2=number of remaining tries (3 at the most)

Command

CLA	INS	P1	P2	P3	UserPin
00	20	00	00	04	3030303030

Response

SW1	SW2	Comment
90	00	Success
63	Number of remaining tries	Fail
67	00	Wrong Length
6B	00	Wrong P1P2

The Verify UserPin2 Command

This command verifies the second user pin. The UserPin2 is required for the memory reading and writing operations.
Upon success it returns the status word SW1 SW2 = 9000
Otherwise it returns SW1=63, SW2=number of remaining tries (3 at the most)

Command

CLA	INS	P1	P2	P3	UserPin2
00	20	00	02	04	3030303030

Response

SW1	SW2	Comment
90	00	Success
63	Number of remaining tries	Fail
67	00	Wrong Length
6B	00	Wrong P1P2

The Verify AdminPin command

This command verifies the administrator pin. It gives access to all available features of the crypto currency application. If P2 is set to FF UserPin and UserPin2 are reset to the default value (four zeros).

Upon success it returns the status word SW1 SW2 = 9000
Otherwise it returns SW1=63, SW2=number of remaining tries (ten at the most)

Command

CLA	INS	P1	P2	P3	AdminPin
00	20	00	01 FF Reset to default	08	30303030303030303030

Response

SW1	SW2	Comment
90	00	Success
63	Number of remaining tries	Fail
67	00	Wrong Length
6B	00	Wrong P1P2

The ChangePIN command

This command sets a PIN (UserPin, UserPin2, AdminPin) to a new value
The P2 value is respectively 00, 02, 01 for UserPin, UserPin2, AdminPin.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=number of remaining tries.

Command

CLA	INS	P1	P2	P3	OldPin	NewPIN
00	24	00	00	10	30303030FFFFFFFF	31313131FFFFFFFF
00	24	00	02	10	30303030FFFFFFFF	30303030FFFFFFFF
00	24	00	01	10	3030303030303030	3131313131313131

Response

SW1	SW2	Comment
90	00	Success
63	Number of remaining tries	Fail
67	00	Wrong Length
6B	00	Wrong P1P2

The GetStatus command

This command returns the current state of the crypto currency application. It required at least the previous checking of one PIN (UserPin, UserPin2, AdminPin).

Command

CLA	INS	P1	P2	P3
00	87	00	00	06

Response

SW1	SW2	Comment
90	00	Success
63	80	PIN required

This command returns 6 bytes.

byte0: Should be 01 (ECDSA signature available)
byte1: The maximum number of keys that can be used by the crypto currency application (08)
byte2byte3: 16 bits (b15...b1b0) indicating the index (bi) of defined keys, for example 0003 for key1 (bit1) and key0 (bit0)
byte4byte5: The size of the user memory (for example 0C00 for 3 KB)

Example

```
// scheck.txt
verbose 1
start
// select
apdu 00A4040006 010203040500
// UserPin= 0000
apdu 0020 0000 04 30303030
// Get Status
apdu 0087 0000 06
```

btools -script scheck.txt

```
Opening the APDU script scheck.txt
Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 87 00 00 06
Rx: 01 08 00 03 0C 00 90 00
```

The Write Command

This command writes data in the non volatile memory. This service could be used for the secure storage of any information in the area [0000, 0C00[.

It requires the previous checking of PINs UserPin2 or AdminPin.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	Data
00	D0	AdrMSB	AdrLSB	Data Length	Data to be written

The starting address ranging from 0000 to 10FF is encoded by two bytes (P1, P2), P1 being the *most significant byte* (MSB) and P2 the *less significant byte* (LSB).

Address Mapping

Start Address	Length	Comment	PIN required
0000	0C00	Data Area	User2 or Admin
0C00	0400	Key Dump Area	Admin
1000	0100	Key Label - 32 bytes/key	Admin

Response

SW1	SW2	comment
90	00	OK
63	80	PIN required
6D	02	Invalid Address

The Read Command

This command reads data in the non volatile memory.
It requires the previous checking of PINs UserPin2 or AdminPin.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3
00	B0	AdrMSB	AdrLSB	Length to be read

The starting address ranging from 0000 to 10FF is encoded by two bytes (P1, P2), P1 being the most significant byte (MSB) and P2 the less significant byte (LSB).

Address Mapping

Start Address	Length	Comment	PIN required
0000	0C00	Data Area	User2 or Admin
0C00	0400	Key Dump Area	Admin
1000	0100	Key Label - 32 bytes/key	Admin

Response

Body	SW1	SW2	comment
Data	90	00	OK
Empty	63	80	PIN required
Empy	6D	01	Invalid Address

The Clear KeyPair Command

This command MUST be used before any key setting or key generation operation.
It requires the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	81	00	Key Index [0,7]	00	Admin

Response

SW1	SW2	Comment
90	00	Key Reset Done
63	80	PIN required
69	85	Bad index

The InitCurve Command

This command initializes the elliptic curve parameters. The keys MUST be cleared before this operation.

It requires the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	89	00 - SECP256k1	Key Index [0,7]	00	Admin
00	89	01 - SECP256v1	Key Index [0,7]	00	Admin

Response

SW1	SW2	Comment
90	00	Key Reset Done
63	80	PIN required
69	85	Bad index
64	01	Public Key is defined
64	02	Private Key is defined
6A	86	Incorrect P1P2

The Generate KeyPair Command

This command generates the elliptic curve public and private keys. The keys MUST be cleared before this operation.

It requires the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	82	00	Key Index [0,7]	00	Admin

Response

SW1	SW2	Comment
90	00	OK
63	80	PIN required
69	85	Bad index
64	01	Public Key is defined
64	02	Private Key is defined
6D	10	Key Generation Error

The Dump KeyPair Command

This command dumps the elliptic curve public and private keys.
It returns the size of the data written in the non volatile memory, in the KeyDump area whose address starts at 0C00

It requires the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	83	00	Key Index [0,7]	02	Admin
00	83	FF Reset DUMP Area	Not Used	00	Admin

Response

Body	SW1	SW2	Comment
Total Length 2 bytes SECP256k1 0185 SECP256v1 0201	90	00	Key Reset Done
	63	80	PIN required
	69	85	Bad index
	64	01	Public Key is not defined
	64	02	Private Key is not defined
	6A	86	Incorrect P1P2

Dump KeysPair: Memory Mapping

Address = 0C00	Comment
Total Length, 2 bytes	
PubKey A Length, 2 bytes	The length of the A parameter
PubKey A parameter value	The value of the A parameter
PubKey B Length, 2 bytes	The length of the B parameter
PubKey B parameter value	The value of the B parameter
PubKey G Length, 2 bytes	The length of the Generator
PubKey G value	The value of the Generator
PubKey R Length, 2 bytes	The length of the R parameter
PubKey R value	The value of the R parameter (Order of the Generator)
PubKey W Length, 2 bytes	The length of the W parameter
PubKey W value	The value of the W Public Key (EC Point)
PubKey Field Length, 2 bytes	The length of the Field parameter
PubKey Field Value	The value of the prime p of the field Z/pZ
PubKey Size, 2 bytes	The size of the Public Key object
PrivKey A Length, 2 bytes	The length of the A parameter
PrivKey A parameter value	The value of the A parameter
PrivKey B Length, 2 bytes	The length of the B parameter
PrivKey B parameter value	The value of the B parameter
PrivKey G Length, 2 bytes	The length of the Generator
PrivKey G value	The value of the Generator
PrivKey R Length, 2 bytes	The length of the R parameter
Privy R value	The value of the R parameter (Order of the Generator)
PrivKey S Length, 2 bytes	The length of the S parameter
PrivKey S value	The value of the S, Private Key (32 bytes)
PrivKey Field Length, 2 bytes	The length of the Field parameter
PrivKey Field Value	The value of the prime p of the field (Z/pZ)
PrivKey Size, 2 bytes	The size of the Private Key object

The Get KeyParameter Command

This command collects the elliptic curve public and private keys parameters

It requires the User or the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	84	00 Parameter A	Key Index [0,7]	00	Admin or User
		01 Parameter B		00	Admin or User
		02 Parameter Field (Z/pZ)		00	Admin or User
		03 Parameter G (generator)		00	Admin or User
		04 Parameter K (cofactor)		00	Admin or User
		05 Parameter R (order of G)		00	Admin or User
		06 Parameter W (Public Key)		43	Admin or User
		07 Parameter S (Private Key		22	Admin
		08 Key Label		20	Admin or User

Response

Body	SW1	SW2	Comment
Param0 Length - Param0 value Param1 Length - Param1 value Param2 Length - Param2 value Param3 Length - Param3 value Param4 Length - Param4 value Param5 Length - Param5 value Param6 Length - Param6 value Param7 Length - Param7 value	90	00	Response includes a 2 bytes length field for parameters 0 to 7
Param8 (Key Label) Value	90	00	OK
	63	80	PIN required
	69	85	Bad index
	64	01	Public Key is not defined
	64	02	Private Key is not defined
	6A	86	Incorrect P1P2
	6D	30	Javacard Exception

The Set KeyParameter Command

This command sets the elliptic curve parameters, including public and private keys parameters

It requires the Admin PIN.
Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	PIN required
00	88	00 Parameter A	Key Index [0,7]	Length	Admin
		01 Parameter B		Length	Admin
		02 Parameter Field (Z/pZ)		Length	Admin
		03 Parameter G (generator)		Length	Admin
		04 Parameter K (cofactor)		Length	Admin
		05 Parameter R (order of G)		Length	Admin
		06 Parameter W (Public Key)		41	Admin
		07 Parameter S (Private Key		20	Admin
		08 Key Label		20	Admin

Response

SW1	SW2	Comment
90	00	OK
63	80	PIN required
69	85	Bad index
64	01	Public Key is defined
64	02	Private Key is defined
6A	86	Incorrect P1P2
6D	40	Javacard Exception

The SignECDSA command

This command generates an ECDSA signature.

It requires the AdminPin or UserPin.
It cancels the UserPin, i.e. a ***UserPin must be presented for every signature***.

Upon success it returns the status word SW1,SW2 = 9000.
Otherwise it returns SW1=63, SW2=80 (PIN required).

Command

CLA	INS	P1	P2	P3	Data	PIN required
00	80	00 Signature without digest	Key Index [0,7]	Length 20	Data to be signed	Admin or User
00	80	21 Signature with SHA256	Key Index [0,7]	Length	Data to be hashed and signed	Admin or User

Response

Body	SW1	SW2	Comment
Length (2 bytes) ASN.1 ECDSA Signature Encoding	90	00	OK
	63	80	PIN required
	69	85	Bad index
	64	01	Public Key is not defined
	64	02	Private Key is not defined
	6A	86	Incorrect P1P2
	6D	20	Signature Error

Annexes

PINs Test Script

```
// s_test_pins.txt

verbose 1
test
start

// select
apdu 00 A4 04 00 06 01 02 03 04 05 00

// Write
apdu 00D0 0000 01 A5
// Read
apdu 00B0 0000 01
// Get Status
apdu 0087 0000 06
// ClearKeys 0
apdu 0081 00 00 00
// InitCurve 0
apdu 0089 00 00 00
// SetParameter 6 (PublicKey 0)
apdu 0088 0600 41
04A6FC0C5F467C3DB8C1581805E7C62C5FAEA19063B01F5845AD68DE9D84385F321EBF3A26B29912418
992DCDC1FE69C282EFF65860E109F53AD27A29624984B6A
// GetParameter 6 (PublicKey 0)
apdu 0088 0600 20  17439172198780D8635D1B2259FF35AC8AACCAE56D914D20C9B9D19D20D029DB

// Verify Admin PIN
apdu 0020 0001 08 31 30 30 30 30 30 30 30
apdu 0020 0001 08 30 30 30 30 30 30 30 30

// Verify User PIN
apdu 0020 0000 04 31 30 30 30
apdu 0020 0000 04 30 30 30 30

// Verify User2 PIN
apdu 0020 0002 04 31 30 30 30
apdu 0020 0002 04 30 30 30 30

// Change User PIN
apdu 0024 0000 10 30 30 30 30 FF FF FF FF 31 31 31 31 FF FF FF FF
// Test User PIN
apdu 0020 0000 04 31 31 31 31
// Block User PIN (3 wrong tries)
apdu 0020 0000 04 30 30 30 30
apdu 0020 0000 04 30 30 30 30
apdu 0020 0000 04 30 30 30 30
// User PIN is blocked
apdu 0020 0000 04 31 31 31 31

// Change User2 PIN
apdu 0024 0002 10 30 30 30 30 FF FF FF FF 31 31 31 31 FF FF FF FF
// Test User2 PIN
apdu 0020 0002 04 31 31 31 31
// Block User2 PIN (3 wrong tries)
apdu 0020 0002 04 30 30 30 30
apdu 0020 0002 04 30 30 30 30
0020 0002 04 30 30 30 30
// User2 PIN is blocked
apdu 0020 0002 04 31 31 31 31

// Verify Admin PIN
apdu 0020 0001 08 30 30 30 30 30 30 30 30
// User PIN is unblocked
```

```
apdu 0020 0000 04 31 31 31 31
// // User2 PIN is unblocked
apdu 0020 0002 04 31 31 31 31

// Verify Admin PIN and reset to default values User and User2 PINs
apdu 0020 00FF 08 30 30 30 30 30 30 30 30

// Verify User   PIN
apdu 0020 0000 04 30 30 30 30
// Verify User2 PIN
apdu 0020 0002 04 30 30 30 30

// Change Admin PIN
apdu 0024 0001 10 30 30 30 30 30 30 30 30 31 30 30 30 30 30 30 30
// Verify Admin Pin, bad value
apdu 0020 0001 08 30 30 30 30 30 30 30 30
// Wong Admin PIN, good value
apdu 0020 0001 08 31 30 30 30 30 30 30 30
// Change Admin PIN
apdu 0024 0001 10 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
// Verify Admin PIN
apdu 0020 0001 08 30 30 30 30 30 30 30 30
```

btools -script s_test_pins.txt

```
Opening the APDU script s_test_pins.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 D0 00 00 01 A5
Rx: 63 80
Tx: 00 B0 00 00 01
Rx: 63 80
Tx: 00 87 00 00 06
Rx: 63 80
Tx: 00 81 00 00 00
Rx: 63 80
Tx: 00 89 00 00 00
Rx: 63 80
Tx: 00 88 06 00 41 04 A6 FC 0C 5F 46 7C 3D B8 C1 58
    18 05 E7 C6 2C 5F AE A1 90 63 B0 1F 58 45 AD 68
    DE 9D 84 38 5F 32 1E BF 3A 26 B2 99 12 41 89 92
    DC DC 1F E6 9C 28 2E FF 65 86 0E 10 9F 53 AD 27
    A2 96 24 98 4B 6A
Rx: 63 80
Tx: 00 88 06 00 20 17 43 91 72 19 87 80 D8 63 5D 1B
    22 59 FF 35 AC 8A AC CA E5 6D 91 4D 20 C9 B9 D1
    9D 20 D0 29 DB
Rx: 63 80
Tx: 00 20 00 01 08 31 30 30 30 30 30 30 30
Rx: 63 09
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 20 00 00 04 31 30 30 30
Rx: 63 02
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 20 00 02 04 31 30 30 30
Rx: 63 02
Tx: 00 20 00 02 04 30 30 30 30
Rx: 90 00
Tx: 00 24 00 00 10 30 30 30 30 FF FF FF FF 31 31 31
    31 FF FF FF FF
Rx: 90 00
```

```
Tx: 00 20 00 00 04 31 31 31 31
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 63 02
Tx: 00 20 00 00 04 30 30 30 30
Rx: 63 01
Tx: 00 20 00 00 04 30 30 30 30
Rx: 63 00
Tx: 00 20 00 00 04 31 31 31 31
Rx: 63 00
Tx: 00 24 00 02 10 30 30 30 30 FF FF FF FF 31 31 31
    31 FF FF FF FF
Rx: 90 00
Tx: 00 20 00 02 04 31 31 31 31
Rx: 90 00
Tx: 00 20 00 02 04 30 30 30 30
Rx: 63 02
Tx: 00 20 00 02 04 30 30 30 30
Rx: 63 01
Tx: 00 20 00 02 04 31 31 31 31
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 20 00 00 04 31 31 31 31
Rx: 90 00
Tx: 00 20 00 02 04 31 31 31 31
Rx: 90 00
Tx: 00 20 00 FF 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 20 00 00 04 30 30 30 30
Rx: 90 00
Tx: 00 20 00 02 04 30 30 30 30
Rx: 90 00
Tx: 00 24 00 01 10 30 30 30 30 30 30 30 30 31 30 30
    30 30 30 30 30
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 63 09
Tx: 00 20 00 01 08 31 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 24 00 01 10 31 30 30 30 30 30 30 30 30 30 30
    30 30 30 30 30
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
```

Read/Write Test Script

```
// s_test_rw.txt

verbose 1
test
start

// Select AID
apdu 00 A4 04 00 06 01 02 03 04 05 00

// Write
apdu 00D0 0000 01 A5
// Read
apdu 00B0 0000 01

// Verify Admin PIN
apdu 0020 0001 08 30 30 30 30 30 30 30 30

// Write
```

```
apdu 00D0 0000 01 A5
// Read
apdu 00B0 0000 01
// Write
apdu 00D0 0C00  01 A5
// Read
apdu 00B0 0C00  10

// select AID
apdu 00 A4 04 00 06 01 02 03 04 05 00

// Write
apdu 00D0 0000 01 A5
// Read
apdu 00B0 0000 01
// Verify User2 PIN
apdu 0020 0002 04 30 30 30 30

// Write
apdu 00D0 0000 01 A5
// Read
apdu 00B0 0000 01
// Read
apdu 00B0 0B80 80

// Write
apdu 00D0 0C00 01 A5
// Read
apdu 00B0 0B80 81
```

btools -script s_test_rw.txt

```
Opening the APDU script s_test_rw.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 D0 00 00 01 A5
Rx: 63 80
Tx: 00 B0 00 00 01
Rx: 63 80
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
Rx: 90 00
Tx: 00 D0 00 00 01 A5
Rx: 90 00
Tx: 00 B0 00 00 01
Rx: A5 90 00
Tx: 00 D0 0C 00 01 A5
Rx: 90 00
Tx: 00 B0 0C 00 10
Rx: A5 85 00 01 00 00 01 07 00 41 04 79 BE 66 7E F9
    90 00
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 D0 00 00 01 A5
Rx: 63 80
Tx: 00 B0 00 00 01
Rx: 63 80
Tx: 00 20 00 02 04 30 30 30 30
Rx: 90 00
Tx: 00 D0 00 00 01 A5
Rx: 90 00
Tx: 00 B0 00 00 01
Rx: A5 90 00
Tx: 00 B0 0B 80 80
Rx: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    90 00
Tx: 00 D0 0C 00 01 A5
Rx: 6D 01
Tx: 00 B0 0B 80 81
Rx: 6D 01
```

ECDSA Test Script

```
// s_test_ecdsa.txt

verbose 1
test 1
start

// select
apdu 00 A4 04 00 06 01 02 03 04 05 00
// Verify PinAdmin
apdu 0020 0001 08 30 30 30 30 30 30 30 30


// Curve 0 SECP256k1
// Keys Generation

// ClearKeys Key0
apdu 0081 00 00 00
// InitCurve 0, Key0
apdu 0089 00 00 00
// GenerateKeysPair Key0
apdu 0082 00 00 00
// Dump KeysPair Key0
apdu 0083 00 00 02
// GetPublicKey Key0
apdu 0084 06 00 00
// GetPrivateKey Key0
apdu 0084 07 00 00

// Curve 0 SECP256k1
// Keys Setting

// ClearKeys Key0
apdu 0081 00 00 00
// InitCurve 0, Key0
apdu 0089 00 00 00
// SetPublicKey Key0
apdu 0088 0600 41
04A6FC0C5F467C3DB8C1581805E7C62C5FAEA19063B01F5845AD68DE9D84385F321EBF3A26B29912418
992DCDC1FE69C282EFF65860E109F53AD27A29624984B6A
// SetPrivateKey Key0
apdu 0088 0700 20  17439172198780D8635D1B2259FF35AC8AACCAE56D914D20C9B9D19D20D029DB


// Signature Generation

// Sign ECDSA Key0
apdu 0080 0000 20 0102030405060708010203040506070801020304050607080102030405060708
// Sign ECDSA-SHA256 Key0
apdu 0080 2100 40
0102030405060708010203040506070801020304050607080102030405060708010203040506070801
2030405060708010203040506070801020304050607080
```

```
// Curve 1- SECP256r1

// ClearKeys Key1
apdu 0081 00 01 00
// InitCurve 1, key1
apdu 0089 01 01 00
// GenerateKeysPair Key1
apdu 0082 00 01 00
// Dump KeysPair Key1
apdu 0083 00 01 02
// GetPublicKey Key1
apdu 0084 06 01 00
// GetPrivateKey Key1
apdu 0084 07 01 00


// SetKeysParamters


// ClearKeys Key0
apdu 0081 00 00 00
// Init Curve SECP256k1 parameters
apdu 0088 00 00 01 00
apdu 0088 01 00 01 07
apdu 0088 02 00 20 fffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f
apdu 0088 03 00 41
0479be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798483ada7726a3c4655
da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8
apdu 0088 04 00 02 00 01
apdu 0088 05 00 20 fffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
apdu 0088 06 00 41
04A6FC0C5F467C3DB8C1581805E7C62C5FAEA19063B01F5845AD68DE9D84385F321EBF3A26B29912418
992DCDC1FE69C282EFF65860E109F53AD27A29624984B6A
apdu 0088 07 00 20 17439172198780D8635D1B2259FF35AC8AACCAE56D914D20C9B9D19D20D029DB
// Sign ECDSA
apdu 0080 0000 20 0102030405060708010203040506070801020304050607080102030405060708


// ClearKeys Key0
apdu 0081 00 00 00
// Init Curve SECP256r1 parameters
apdu 0088 00 00 20 ffffffff00000001000000000000000000000000ffffffffffffffffffffffffc
apdu 0088 01 00 20 5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
apdu 0088 02 00 20 ffffffff00000001000000000000000000000000ffffffffffffffffffffffff
apdu 0088 03 00 41
046b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c2964fe342e2fe1a7f9b8
ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5
apdu 0088 04 00 02 00 01
apdu 0088 05 00 20 ffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551
apdu 0088 06 00 41
047C46A3FCCC92AE3925105E45B46F64B770F76C96539797833C2AEE8C7FE6BFEF661BBB78B654A434E
58993BB35AD8D4AD73739F3D48B515CAAA02D7826765299
apdu 0088 07 00 20 28DACF212405D69221C11E9DB7C52A8F33CB9496895A596D2A07CE7D1D2902C8
// Sign ECDSA
apdu 0080 0000 20 0102030405060708010203040506070801020304050607080102030405060708
```

btools -script s_test_ecdsa.txt

```
Opening the APDU script s_test_ecdsa.txt

Reader: Broadcom Corp Contacted SmartCard 0
T=0  - ATR
Tx: 00 A4 04 00 06 01 02 03 04 05 00
Rx: 90 00
Tx: 00 20 00 01 08 30 30 30 30 30 30 30 30
```

```
Rx: 90 00
Tx: 00 81 00 00 00
Rx: 90 00
Tx: 00 89 00 00 00
Rx: 90 00
Tx: 00 82 00 00 00
Rx: 90 00
Tx: 00 83 00 00 02
Rx: 01 85 90 00
Tx: 00 84 06 00 00
Rx: 6C 43
Tx: 00 84 06 00 43
Rx: 00 41 04 BA 5A 71 A8 0E 90 76 9E DD D2 B9 6C B4
    BA 47 0B 45 C6 3B 01 F5 A9 FB FC 3F 95 37 43 23
    18 15 5D 59 F3 F1 75 26 08 4E 5A CC 7D 17 4D 68
    AB 39 57 C4 F6 D8 5D 38 43 95 EF 8D F4 7D 05 3B
    FE E6 F9 90 00
Tx: 00 84 07 00 00
Rx: 6C 22
Tx: 00 84 07 00 22
Rx: 00 20 85 1F 6D 62 0B 87 FC 27 FC 9A 00 42 8F C6
    01 37 D8 6B 14 07 E4 B6 8F 77 30 A4 BF AC CE 7D
    A3 91 90 00
Tx: 00 81 00 00 00
Rx: 90 00
Tx: 00 89 00 00 00
Rx: 90 00
Tx: 00 88 06 00 41 04 A6 FC 0C 5F 46 7C 3D B8 C1 58
    18 05 E7 C6 2C 5F AE A1 90 63 B0 1F 58 45 AD 68
    DE 9D 84 38 5F 32 1E BF 3A 26 B2 99 12 41 89 92
    DC DC 1F E6 9C 28 2E FF 65 86 0E 10 9F 53 AD 27
    A2 96 24 98 4B 6A
Rx: 90 00
Tx: 00 88 07 00 20 17 43 91 72 19 87 80 D8 63 5D 1B
    22 59 FF 35 AC 8A AC CA E5 6D 91 4D 20 C9 B9 D1
    9D 20 D0 29 DB
Rx: 90 00
Tx: 00 80 00 00 20 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08
Rx: 61 49
Tx: 00 C0 00 00 49
Rx: 00 47 30 45 02 21 00 C3 F5 BB F9 58 C5 CC 55 6B
    0A 60 A9 9F 53 A6 54 8F DC B6 90 4C A1 23 25 34
    78 8A 5F ED 59 7E C2 02 20 01 EC 15 95 20 D3 8E
    7F CE AD 5D B7 D9 80 5D 20 88 22 A5 7E B1 2C AC
    8E 24 47 61 5C EA 63 33 24 90 00
Tx: 00 80 21 00 40 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08
Rx: 61 48
Tx: 00 C0 00 00 48
Rx: 00 46 30 44 02 20 1D DC 70 50 F9 75 68 C7 E9 AA
    FF C7 8D E5 DC D5 D2 93 75 FE 18 5F 42 C6 48 A8
    6C 56 2F 29 6E 2A 02 20 25 5C 0C FE A5 AF EB 88
    7E 33 89 8E AC DB 6E 25 93 9B 90 EA 53 53 B8 EC
    F3 B5 32 8E B7 BD DB F7 90 00
Tx: 00 81 00 01 00
Rx: 90 00
Tx: 00 89 01 01 00
Rx: 90 00
Tx: 00 82 00 01 00
Rx: 90 00
Tx: 00 83 00 01 02
Rx: 02 01 90 00
Tx: 00 84 06 01 00
```

```
Rx: 6C 43
Tx: 00 84 06 01 43
Rx: 00 41 04 FA D1 F1 E6 74 D2 B3 4E 20 3A 41 17 AD
    D1 56 DF 9C 43 61 31 FD 6B CE D2 4E AC A4 1E 43
    6B 35 DD B2 87 B3 53 B3 D0 E0 07 CA 68 11 CC 4A
    D8 42 95 3B A5 EE 8D C3 EF E9 BE ED 91 95 99 AB
    3B CE 16 90 00
Tx: 00 84 07 01 00
Rx: 6C 22
Tx: 00 84 07 01 22
Rx: 00 20 CE 88 A1 37 5E BA A9 44 4F DD 26 19 35 4C
    72 5B 75 F3 8C A1 68 8F FD E8 58 3F 41 07 CE 50
    16 7B 90 00
Tx: 00 81 00 00 00
Rx: 90 00
Tx: 00 88 00 00 01 00
Rx: 90 00
Tx: 00 88 01 00 01 07
Rx: 90 00
Tx: 00 88 02 00 20 FF FF FF FF FF FF FF FF FF FF FF
    FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
    FE FF FF FC 2F
Rx: 90 00
Tx: 00 88 03 00 41 04 79 BE 66 7E F9 DC BB AC 55 A0
    62 95 CE 87 0B 07 02 9B FC DB 2D CE 28 D9 59 F2
    81 5B 16 F8 17 98 48 3A DA 77 26 A3 C4 65 5D A4
    FB FC 0E 11 08 A8 FD 17 B4 48 A6 85 54 19 9C 47
    D0 8F FB 10 D4 B8
Rx: 90 00
Tx: 00 88 04 00 02 00 01
Rx: 90 00
Tx: 00 88 05 00 20 FF FF FF FF FF FF FF FF FF FF FF
    FF FF FF FF FE BA AE DC E6 AF 48 A0 3B BF D2 5E
    8C D0 36 41 41
Rx: 90 00
Tx: 00 88 06 00 41 04 A6 FC 0C 5F 46 7C 3D B8 C1 58
    18 05 E7 C6 2C 5F AE A1 90 63 B0 1F 58 45 AD 68
    DE 9D 84 38 5F 32 1E BF 3A 26 B2 99 12 41 89 92
    DC DC 1F E6 9C 28 2E FF 65 86 0E 10 9F 53 AD 27
    A2 96 24 98 4B 6A
Rx: 90 00
Tx: 00 88 07 00 20 17 43 91 72 19 87 80 D8 63 5D 1B
    22 59 FF 35 AC 8A AC CA E5 6D 91 4D 20 C9 B9 D1
    9D 20 D0 29 DB
Rx: 90 00
Tx: 00 80 00 00 20 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08
Rx: 61 49
Tx: 00 C0 00 00 49
Rx: 00 47 30 45 02 21 00 DF 60 40 61 02 23 6B 1A 74
    D9 CB 1F 09 F3 AA A0 7D 44 5C B5 25 C7 46 60 D3
    68 DB 44 46 52 B5 47 02 20 32 6C 23 AB 25 EC A3
    9D 38 15 99 D9 50 43 64 E5 68 A8 75 AF 09 AF 78
    70 F9 A5 3D 07 32 F8 8D F1 90 00
Tx: 00 81 00 00 00
Rx: 90 00
Tx: 00 88 00 00 20 FF FF FF FF 00 00 00 01 00 00 00
    00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF
    FF FF FF FF FC
Rx: 90 00
Tx: 00 88 01 00 20 5A C6 35 D8 AA 3A 93 E7 B3 EB BD
    55 76 98 86 BC 65 1D 06 B0 CC 53 B0 F6 3B CE 3C
    3E 27 D2 60 4B
Rx: 90 00
Tx: 00 88 02 00 20 FF FF FF FF 00 00 00 01 00 00 00
    00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF
    FF FF FF FF FF
```

```
Rx: 90 00
Tx: 00 88 03 00 41 04 6B 17 D1 F2 E1 2C 42 47 F8 BC
    E6 E5 63 A4 40 F2 77 03 7D 81 2D EB 33 A0 F4 A1
    39 45 D8 98 C2 96 4F E3 42 E2 FE 1A 7F 9B 8E E7
    EB 4A 7C 0F 9E 16 2B CE 33 57 6B 31 5E CE CB B6
    40 68 37 BF 51 F5
Rx: 90 00
Tx: 00 88 04 00 02 00 01
Rx: 90 00
Tx: 00 88 05 00 20 FF FF FF FF 00 00 00 00 FF FF FF
    FF FF FF FF FF BC E6 FA AD A7 17 9E 84 F3 B9 CA
    C2 FC 63 25 51
Rx: 90 00
Tx: 00 88 06 00 41 04 7C 46 A3 FC CC 92 AE 39 25 10
    5E 45 B4 6F 64 B7 70 F7 6C 96 53 97 97 83 3C 2A
    EE 8C 7F E6 BF EF 66 1B BB 78 B6 54 A4 34 E5 89
    93 BB 35 AD 8D 4A D7 37 39 F3 D4 8B 51 5C AA A0
    2D 78 26 76 52 99
Rx: 90 00
Tx: 00 88 07 00 20 28 DA CF 21 24 05 D6 92 21 C1 1E
    9D B7 C5 2A 8F 33 CB 94 96 89 5A 59 6D 2A 07 CE
    7D 1D 29 02 C8
Rx: 90 00
Tx: 00 80 00 00 20 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08 01 02 03 04 05 06 07 08 01 02 03
    04 05 06 07 08
Rx: 61 48
Tx: 00 C0 00 00 48
Rx: 00 46 30 44 02 20 6D E4 BD F1 12 73 FD 45 22 F5
    68 62 C9 9C DA 63 A6 7E 79 6A 1C 63 6E 4A F2 B3
    13 0D CE ED 63 7F 02 20 71 50 C6 D2 CB A5 78 35
    19 CA 26 B8 F1 B1 15 00 D4 CE 92 39 DA A8 DA 54
    1A 0C 72 40 F9 FD FD 06 90 00
```