Asymmetic Keys

Two keys one public and one private.

- □□
- □□: Public Key
- □□: Private Key
- Encrypt: □□□□
- Decrypt: □□ □□
- Cipher□□is non-commutative
- ?

?

□□ is a mighty operator, function and a cloud data source all at once!

As an operator or function it is used with no arguments and it creates the Free Form Programming Language's (ff) default blockchain whic is currently set to Bloxberg 2.0.

In the ff script below you can see on the first line □□ acts as singleton operator or function and , and on the 3rd line as a cloud data source:

```
□□

show □□

save as blox;
```

Output

"□□" → {"type" -> "\"elliptic\"", "curve" -> "\"ethereum\"", "compressed" -> "False"}

```
□□

tmp = "□□curve";

show tmp;

save as blox;
```

Output

"tmp" → ethereum

?: Public Key

?: Private Key

Encrypt: ??

- Symbol □□behaves like a binary operator
- □□ encrypts what is on its rhs e.g. □□m
- □□□□uses the key on its lhs to encrypt
- crypt = □□□□ m just like e.g. multiplication copies the resultant into the lhs of =

> □□□□ m behaves like a product by the operator □□incorporating is rhs and lhs into a product.

```
□□["rsa"] ;

m = "hi";

crypt = □□□□m;

show crypt;

save as rsa;
```

Output:

"crypt" ⟶ "<|

\"Cipher\" -> \"RSA\",

\"Data\" ->
ByteArray[\"UK0fWzfMEKdQ+aQnc5a3BX0C7ptg4aEa5mbXQhIRz+/17WVRIQ+atsjSET8Rin7BsIPaT
W851pky8dbTLNNg8vja0mI572KZJXRM9YGaFte2UkqwGl4OSEdmv+fXD7KbVi0Ps/uO/EskrksneIPSCn
IMaJ8VkSRjcNsXjuAoPvJ92mwvvfSSdMNxm0FJjuCMSgGZRfwwEK0XWviKcmV4eljBv2m9NT4Rk1e130
EXovnqtGS5XNyGzDQew8fHtgRreeTmSwOO7HCfrqmzYH14aBTRf+KuLLcaPpKo98PGeCmnDW56Cp
DWHsnYsOtkq/oY/9ACHwPk9hR5Hfl9O2l2Yg==\"],

\"OriginalForm\" -> String,

\"Padding\" -> \"PKCS1\"

|>"

Decrypt: ? ?

```
□□["rsa"] ;

bob = "hi";

crypt  = □□□□ob;

alice = crypt □□□□;

show bob also alice;

save as rsa;
```

Output:

"bob" ⟶ "(\"hi\")"

"alice" ⟶ "(\"hi\")"

Cipher?is non-commutative

□□□□ encrypts while □□□□ decrypts!

□□is called Cipher.

Cipher□□acts like a binary operator.

?

1. No matter , if the key on the left of the cipher □□ is public and the right side only the private key, how about a single key □ symbol which is replaced according to its corresponding location relative to □□. y = □ □□x encrypts x into y and x = y□□□ decrypts y back to x. If the cipher □□has asymmetric keys this works fine, and symmetric key ciphers by definition fits.