# demo tool for express of trust

community.tokenscript.org/t/demo-tool-for-express-of-trust/235

June 19, 2019

lyhistory

Jun '19

> https://github.com/AlphaWallet/TokenScript/blob/master/doc/security.md 2

based on the design in the doc, I created a demo tool with a small test case in JUnit test for generating proof-of-trust and proof-of-revoke address.

currently, it's hard for me to spare time to work on it as I'm still on a two-week business trip, welcome to contribute your effort to help me improve the tool. and also there is one small question regarding the statement:

> it can calculate the private key of that address by $x{\cdot}h$ given that $(g^x)^h = (g^h)^x$.

at first I thought it's talking about RSA, $Y' = Y^h = (g^x)^h = g^{xh}$ = RSA.generate(Y, $e$=$h$) and then I realised ECC also belongs to *discrete logarithm problem*(still not clear why it belongs to it, it looks to me that ecc is scala multiplicative not exponents, need to take time to find out and please comment below explain to me if you know)

I still feel it's better to use denote like this: h*(x*G)=x*(h*G)

for an elliptic curve E over a prime field Fp (denoted by E(Fp)), elliptic curve belongs to abelian groups, so it satisfies commutative and associative for Point Addition

Y=x*G

Y' = h*Y =h*(x*G) = (x*h)*G

about abelian group:

A set G is called an abelian group (G,* ) with a binary operation *, G*G->G if it satisfies the following properties:

1. Associativity: (a*b)*c=a*(b*c) for all a,b,c in G .
2. Commutativity: a*b=b*a for all a,b in G .
3. Identity: there exists i in G such that i*a=a*i=a, for all a in G .
4. Inverse: for each a in G , there exists b in G such that a*b=b*a=i . Element b is called the inverse of a.

- **created**

  Jun '19
- **last reply**

  Jun '19
- 3

  **replies**
- 197

  **views**
- 2

  **users**
- 2

  **links**
- J

J

jot2re

Jun '19

I completely agree with the issue of the notation. The reasoning for this notation (despite it being unorthodox for elliptic curves) was to increase the readability. A lot of people are not familiar with additive group notation and point addition, whereas the multiplicative notation over the integers are conceptually easier for most people to understand. However, the more I think about it, I also find that this might actually increase confusion. So I agree that we should change the notation in secure.md to additive point notation. Again note that even the fact that we are doing multiplication in an additive group might also seem weird, but the key is to note that addition is defined over points, whereas the multiplication we are actually doing is of a scalar. That is, we are adding the same point to itself multiple times hence we are actually still only doing additions over the elements in the group.

J

jot2re

Jun '19

I had a look at the code and it looks sensible and seems to do what it is supposed to. It still seems to be a bit rough around the edges though, in the sense of out-commented lines, temporary handling of exceptions and limited tests.

lyhistory

Jun '19

Thanks Tore, fixed a small bug and packaged, now the tool can be downloaded here https://github.com/lyhistory/TokenScriptTool/releases/tag/0.0.1 2