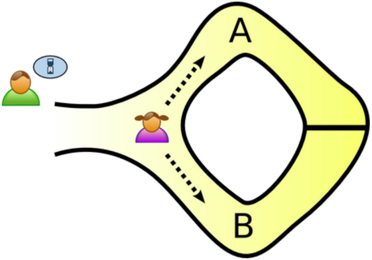
An introduction of zero knowledge proofs

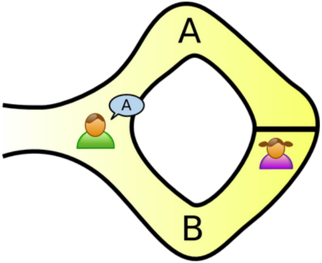
Manta Network

June 2, 2021

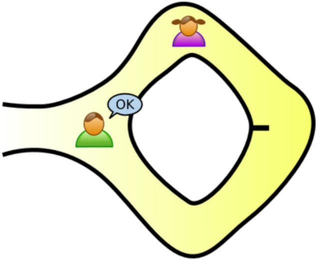
- Prover
- Verifier
- Statement
    - "I know x"
    - "I know $x_1 + x_2 = x_3$"
    - "I know $x_1 \times x_2 = x_3$"
- Property
    - nothing about x, x_1, x_2, x_3 is leaked to Verifier
    - Verifier is convince about the statement



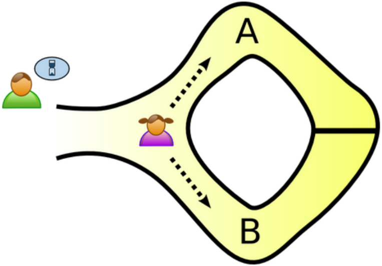
When your friend isn't looking, you go into one side of the cave.[1]



You wait for your friend to tell you which side to come out of.[2]

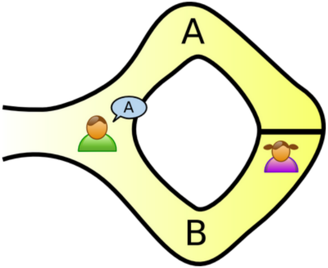


If you succeed enough times, your friend will trust that you know the code.[3]



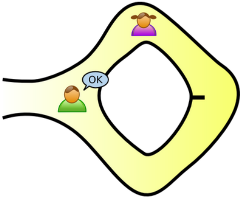
When your friend isn't looking, you go into one side of the cave.[1]

1. Commit to some task



You wait for your friend to tell you which side to come out of.[2]

1. Commit to some task
2. receive a random challenge



If you succeed enough times, your friend will trust that you know the code.[3]

1. Commit to some task
2. receive a random challenge
3. return a reply

Discrete log problem

- Let $\mathbb{G}$ be a cyclic group; let G be a generator of the group.
- Given $X := xG$, find x.

Preliminary

Discrete log problem

- Let $\mathbb{G}$ be a cyclic group; let G be a generator of the group.
- Given $X := xG$, find x.

Example

- $\mathbb{G} = \mathbb{F}_{41}$, $G = 10$.
- $x = 20$, $X = 20 \times 10 \equiv 36 \bmod 41$.
- Given 36 it is (supposedly) hard to find 20

Preliminary

Discrete log problem

- Let $\mathbb{G}$ be a cyclic group; let G be a generator of the group.
- Given $X := xG$, find x.

Real world

- $\mathbb{G}$ is a prime order subgroup of sum elliptic curve (i.e. Curve 25519)
- Cost to find discrete log $\sqrt{|\mathbb{G}|}$

Methodology

1. Commit to some task

Instantiation

1. I know some secret x for some $X := xG$

Methodology

1. Commit to some task
2. receive a random challenge

Instantiation

1. I know some secret x for some $X := xG$
2. Here is a challenge c

Methodology

1. Commit to some task
2. receive a random challenge
3. return a reply

Instantiation

1. I know some secret x for some $X := xG$
2. Here is a challenge c
3. Return $F(c, x)$

Schnorr identification, a.k.a, Σ protocol

$$
\begin{array}{lcr}
\text{Prover } (x, X := xG) & & \text{Verifier } (X) \\
\hline
y \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|},\ Y = yG & & \\
 & \xrightarrow{\quad Y \quad} & \\
 & & c \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} \\
 & \xleftarrow{\quad c \quad} & \\
z = y - cx & & \\
 & \xrightarrow{\quad z \quad} & \\
 & & zG \stackrel{?}{=} Y - cX
\end{array}
$$

Schnorr identification, a.k.a, Σ protocol

$$
\begin{array}{lcr}
\text{Prover } (x, X := xG) & & \text{Verifier } (X) \\
\hline
y \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|},\ Y = yG & & \\
 & \xrightarrow{\quad Y \quad} & \\
 & & c \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} \\
 & \xleftarrow{\quad c \quad} & \\
z = y - cx & & \\
 & \xrightarrow{\quad z \quad} & \\
 & & zG \stackrel{?}{=} Y - cX
\end{array}
$$

Correctness

- $zG = (y - cx)G = yG - cxG = Y - cX$

Schnorr identification, a.k.a, Σ protocol

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_{\$} \mathbb{Z}_{\|\mathbb{G}\|},\ Y = yG$		
	$\xrightarrow{\quad Y \quad}$	
		$c \leftarrow_{\$} \mathbb{Z}_{\|\mathbb{G}\|}$
	$\xleftarrow{\quad c \quad}$	
$z = y - cx$		
	$\xrightarrow{\quad z \quad}$	
		$zG \stackrel{?}{=} Y - cX$

Soundness

- $1/|\mathbb{G}|$ prob that Prover may cheat
- Probabilistic Checkable proof (PCP)

Schnorr identification, a.k.a, Σ protocol

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert},\ Y = yG$		
	$\xrightarrow{\quad Y \quad}$	
		$c \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert}$
	$\xleftarrow{\quad c \quad}$	
$z = y - cx$		
	$\xrightarrow{\quad z \quad}$	
		$zG \stackrel{?}{=} Y - cX$

Unforgability via rewinding (a.k.a., forking lemma)

- Suppose simulator challenges Y on two different c and c'
- Prover returns z and z'
- Then $z - z' = (y - cx) - (y - c'x) = (c' - c)x \rightarrow x = \frac{z-z'}{c'-c}$

None interactive Schnorr identification

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert},\ Y = yG$		
	$\xrightarrow{\quad Y \quad}$	
$c \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert}$		
	$\xrightarrow{\quad c \quad}$	
$z = y - cx$		
	$\xrightarrow{\quad z \quad}$	
		$zG \stackrel{?}{=} Y - cX$

None interactive Schnorr identification

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert},\ Y = yG$		
	$\xrightarrow{\quad Y \quad}$	
$c \leftarrow_{\$} \mathbb{Z}_{\lvert\mathbb{G}\rvert}$		
	$\xrightarrow{\quad c \quad}$	
$z = y - cx$		
	$\xrightarrow{\quad z \quad}$	
		$zG \stackrel{?}{=} Y - cX$

- Prover may cheat on c

None interactive Schnorr identification

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_{\$} \mathbb{Z}_{\|\mathbb{G}\|}$, $Y = yG$		
	$\xrightarrow{\quad Y \quad}$	
$c = Hash(X, Y)$		
	$\xrightarrow{\quad c \quad}$	
$z = y - cx$		
	$\xrightarrow{\quad z \quad}$	
		$zG \stackrel{?}{=} Y - cX$
		$c \stackrel{?}{=} Hash(X, Y)$

None interactive Schnorr identification

Prover $(x, X := xG)$		Verifier (X)
$y \leftarrow_\$ \mathbb{Z}_{\lvert\mathbb{G}\rvert}$, $Y = yG$		
	$\xrightarrow{Y}$	
$c = Hash(X, Y)$		
	$\xrightarrow{c}$	
$z = y - cx$		
	$\xrightarrow{z}$	
		$zG \stackrel{?}{=} Y - cX$
		$c \stackrel{?}{=} Hash(X, Y)$

- Fiat-Shamir Transformation
- $Hash(\cdot)$ is modelled as a random oracle

Statement: "I know x", a.k.a Schnorr signature

- Sign(x, X, msg):
  - $y \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|}$, $Y = yG$
  - $c = hash(msg|X|Y)$
  - $z = y - sc$
  - $\sigma = \{z, c\}$

- Verify(X, msg, σ):
  - $Y' = zG + cX$
  - $c \stackrel{?}{=} hash(msg|X|Y')$

Statement: "I know $x_1 + x_2 = x_3$"

Prover $(x_1, x_2, x_3, X_1, X_2, X_3)$		Verifier (X_1, X_2, X_3)
$y_1, y_2, y_3 \leftarrow_\$ \mathbb{Z}_{\lvert\mathbb{G}\rvert}$		
	$\xrightarrow{Y_1, Y_2, Y_3}$	
	$\xleftarrow{c}$	$c \leftarrow_\$ \mathbb{Z}_{\lvert\mathbb{G}\rvert}$
$z_1 = y_1 - cx_1$ $z_2 = y_2 - cx_2$ $z_3 = y_3 - cx_3$		
	$\xrightarrow{z_1, z_2, z_3}$	
		?

Statement: “I know $x_1 + x_2 = x_3$”

Prover $(x_1, x_2, x_3, X_1, X_2, X_3)$		Verifier (X_1, X_2, X_3)
$y_1, y_2, y_3 \leftarrow_\$ \mathbb{Z}_{\lvert\mathbb{G}\rvert}$		
	$\xrightarrow{Y_1, Y_2, Y_3}$	
	$\xleftarrow{c}$	$c \leftarrow_\$ \mathbb{Z}_{\lvert\mathbb{G}\rvert}$
$z_1 = y_1 - cx_1$ $z_2 = y_2 - cx_2$ $z_3 = y_3 - cx_3$		
	$\xrightarrow{z_1, z_2, z_3}$	
		$z_1 G \stackrel{?}{=} Y_1 - cX_1$ $z_2 G \stackrel{?}{=} Y_2 - cX_2$ $z_3 G \stackrel{?}{=} Y_3 - cX_3$?

Bilinear pairing

- A map $e : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_t$
- let $G_1 \in \mathbb{G}_1$ and $G_2 \in \mathbb{G}_2$, for any $x, y \in \mathbb{Z}$

$$e(xG_1, yG_2) = e(G_1, G_2)^{xy}$$

Prove $x_1 + x_2 = x_3$ is equiv to $e((x_1 + x_2)G_1, G_2) = e(G_1, x_3 G_2)$

- $e((x_1 + x_2)G_1, G_2) = e(G_1, G_2)^{x_1 + x_2}$
- $e(G_1, x_3 G_2) = e(G_1, G_2)^{x_3}$
- $e((x_1 + x_2)G_1, G_2) = e(G_1, x_3 G_2) \rightarrow x_1 + x_2 = x_3$

Statement: "I know $x_1 + x_2 = x_3$"

$$
\begin{array}{ccc}
\text{Prover } (x_1, x_2, x_3) & & \text{Verifier } (X_1, X_2, X_3) \\
X_1 := x_1 G_1, X_2 := x_2 G_1 & & \\
X_3 := x_3 G_2 & & \\
\hline
y_1, y_2, y_3 \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} & & \\
Y_1 = y_1 G_1, y_2 = y_2 G_1 & & \\
Y_3 = y_3 G_2 & & \\
 & \xrightarrow{Y_1, Y_2, Y_3} & \\
 & \xleftarrow{c} & c \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} \\
z_1 = y_1 - cx_1 & & \\
z_2 = y_2 - cx_2 & & \\
z_3 = y_3 - cx_3 & & \\
 & \xrightarrow{z_1, z_2, z_3} & \\
 & & z_1 G_1 \stackrel{?}{=} Y_1 - cX_1 \\
 & & z_2 G_1 \stackrel{?}{=} Y_2 - cX_2 \\
 & & z_3 G_2 \stackrel{?}{=} Y_3 - cX_3 \\
 & & e(X_1 + X_2, G_2) \stackrel{?}{=} e(G_1, X_3)
\end{array}
$$

Statement: "I know $x_1 \times x_2 = x_3$"

$$
\begin{array}{ccc}
\text{Prover } (x_1, x_2, x_3) & & \text{Verifier } (X_1, X_2, X_3) \\
X_1 := x_1 G_1, X_2 := x_2 G_2 & & \\
X_3 := x_3 G_1 & & \\
\hline
y_1, y_2, y_3 \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} & & \\
Y_1 = y_1 G_1, y_2 = y_2 G_2 & & \\
Y_3 = y_3 G_1 & & \\
& \xrightarrow{\quad Y_1, Y_2, Y_3 \quad} & \\
& \xleftarrow{\quad c \quad} & c \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|} \\
z_1 = y_1 - cx_1 & & \\
z_2 = y_2 - cx_2 & & \\
z_3 = y_3 - cx_3 & & \\
& \xrightarrow{\quad z_1, z_2, z_3 \quad} & \\
& & z_1 G_1 \stackrel{?}{=} Y_1 - cX_1 \\
& & z_2 G_2 \stackrel{?}{=} Y_2 - cX_2 \\
& & z_3 G_1 \stackrel{?}{=} Y_3 - cX_3 \\
& & e(X_1, X_2) \stackrel{?}{=} e(X_3, G_2)
\end{array}
$$

Backup slides: VRF

"Double" Schnorr identification

Sender $(x, X := xG)$		Receiver (X)
$y \leftarrow_{\$} \mathbb{Z}_{\|\mathbb{G}\|},\ Y_1 = yG$		
$H = hash(T),\ Y_2 = yH,\ V = xH$		
	$\xrightarrow{Y_1, Y_2, V}$	
		$c \leftarrow_{\$} \mathbb{Z}_{\|\mathbb{G}\|}$
	$\xleftarrow{c}$	
$z = y - cx$		
	$\xrightarrow{z}$	
		$H = hash(T)$
		$zG \stackrel{?}{=} Y_1 - cX$
		$zH \stackrel{?}{=} Y_2 - cV$

ECVRF

- Prove(x, X, msg):
  - $y \leftarrow_{\$} \mathbb{Z}_{|\mathbb{G}|}$, $Y_1 = rG$
  - $H = hash_1(msg|pk)$, $Y_2 = rH$, $V = sH$
  - $c = hash_2(msg|H|X|Y_1|Y_2)$
  - $z = y - cx$
  - $\sigma = \{z, c\}$, $\pi = V$

- Verify(X, msg, σ, π):
  - $Y_1' = zG + cX$
  - $H' = hash_1(msg|X)$
  - $Y_2' = zH' + cV$
  - $c \stackrel{?}{=} hash_2(msg|H'|X|Y_1'|Y_2')$

Security requirements

- Correctness
- Unforgability follows Schnorr signature
- Uniqueness: fix H, msg, there is only one V
- Pseudorandomness: V is IND from random

Uniqueness in ECVRF

Suppose $\{z, c, V\}$ is a valid ECVRF for *msg* and *pk*

$$
\begin{aligned}
& Y_1' = zG + cX, \quad Y_2' = zH + cV \\
\implies \quad & Y_1'/G = z + cx, \quad Y_2'/H = z + cV/H, \\
\implies \quad & Y_1'/G - Y_2'/B = c(x - V/H) \\
\implies \quad & c = \frac{Y_1'/G - Y_2'/H}{x - V/H}
\end{aligned}
$$

OTOH,

$$c = hash(msg|H|xG|Y_1'|Y_2')$$

c is uniquely defined by the input to the RO.