Some Notes

Brandon Goodell, Rigo Salazar, Freeman Slaughter, Luke Szramowski

Cypher Stack

May 24, 2025

1 Introduction

Feldman's (n,t) Verifiable Secret Sharing Scheme		
Input: public G		
Dealer		**Party P_i**
Sample $a_0, \ldots, a_t \xleftarrow{\$} \mathbb{F}_p^*$		
Form $f(x) = a_0 + a_1 x + \cdots + a_t x^t$ for secret a_0		
Share commits $A_k = a_k G$ for $i = 0, \ldots, t$	$\xrightarrow{(A_0,\ldots,A_t)}$	
Share $s_i = f(i)$	$\xrightarrow{s_i}$	
		P_i uses $(A_0, \ldots, A_t)$ to verify s_i is valid
		Set $V = s_i G$ and $V' = A_0 + iA_1 + \cdots + i^t A_t$
		Define $S = V' - V$. Normally, check that $S = 0$
		Eagen/Bassa stuff: $\sum_{k=0}^{t} \frac{i^k}{x-\pi(A_k)} - \frac{s_i}{x-\pi(G)}$

Table 1: Feldman's protocol, with the Sum of Points stuff.

Point is, we can cut down on P_i's point computations from $t+1$ to 0, permitting some error. The secret a_0 will eventually be recovered using Lagrange interpolation - can we also use divisor stuff for this??

An Actual Proof of Knowledge		
Public: G, $B_i = 2^i G$ **for** $i = 0, \ldots, k$		
Private: s_i **such that** $a = s_0 + 2s_1 + \cdots + 2^k s_k$		
Prover		**Verifier**
Sample $r \xleftarrow{\$} \mathbb{F}_p^*$		
Factor $r = r_0 + 2r_1 + \cdots + 2^k r_k$		
Set $Q = rG$	$\xrightarrow{Q}$	
	$\xleftarrow{e}$	Sample $e \xleftarrow{\$} \mathbb{F}_p^*$
Form $c_i = s_i + er_i$ for $i = 0, \ldots, k$	$\xrightarrow{c_0,\ldots,c_k}$	
		Check $c_0 B_0 + \cdots + c_k B_k \stackrel{?}{=} P + eQ$
		Use divisor nonsense for *this* check!

Table 2:

This protocol is complete due to the computation:

$$
\begin{aligned}
& c_0B_0 + \cdots + c_kB_k \\
& = (s_0 + er_0)G + \cdots + (s_k + er_k)(2^kG) \\
& = (s_0 + \cdots + 2^ks_k)G + e(r_0 + \cdots + 2^kr_k)G \\
& = P + eQ
\end{aligned}
$$