Note on Paying for a Pedersen Commitment Opening

Lloyd Fournier (lloyd.fourn@gmail.com)

December 2019

1 Creating The Access Structure

In this short note we demonstrate how a to securely sell an opening (r, x) to a Pedersen Commitment $C = rG + xH$ on Bitcoin. The protocol begins with the seller publishing a simple discrete log access structure for the opening. If the buyer is able to learn the discrete logarithms of two points (A and B below) the they will be able to decrypt the opening.

Seller(r, x)		**Buyer**
$(a, b) \leftarrow_{\$} \mathbb{Z}_q$		
$A \leftarrow aG; B \leftarrow bG; B' \leftarrow bH$		
$\alpha \leftarrow r + a; \beta \leftarrow x + b$		
$\pi \leftarrow \mathsf{P}_{\mathsf{DLEQ}}((G, B), (H, B'), b)$		
	$\xrightarrow{\alpha, \beta, A, B, B', \pi}$	
		$\mathsf{V}_{\mathsf{DLEQ}}((G, B), (H, B'), \pi) \stackrel{?}{=} 1$
		$C + A + B' \stackrel{?}{=} \alpha G + \beta H$
return $((a, A), (b, B))$		**return** (A, B, α, β)

Fig. 1. The access structure setup protocol. $(\mathsf{P}_{\mathsf{DLEQ}}, \mathsf{V}_{\mathsf{DLEQ}})$ denote the non-interactive proving and verification algorithms for discrete logarithm equality [1]

1.1 Security

Correctness follows from the fact that $C = rG + xH = \alpha G + \beta H - A - B'$. If the buyer learns a and b such that $A = aG$ and $B = bG$, then $(r = \alpha - a, x = \beta - b)$ must be a valid opening for C (if the proof π is sound). We must also ensure that the buyer learns nothing about the opening until they learn a and b. This follows from the fact that a valid looking tuple $(\alpha, \beta, A, B, B', \pi)$ is *simulatable* i.e. it can be produced without knowledge of the opening of C: choose b, α, β as normal and then set $A \leftarrow \alpha G + \beta H - B' - C$.

2 Purchasing the dicrete logaritihms

The buyer with (A, B, α, β), can generate a valid opening for C given (a,b), the discrete logarithms of (A, B). Thus, the buyer now attempts to purchase (a, b) from the seller. In case, the seller does not know (a, b) the buyer must have their money returned. Thus they construct the following transaction scaffold

1. Fund: Spends from the buyers inputs with two outputs whose value adds up to v.
2. Redeem: Spends the two outputs of Fund to the seller's address
3. Refund: Spends the two outputs of Fund to the buyer's address (time-locked)

To complete the scaffold they define transitions between transactions in the scaffold by exchanging signatures on the Redeem and Refund transactions as follows:

1. They jointly sign both inputs of the Refund transaction such that the buyer has a valid witness for it.
2. They jointly produce two one-time encrypted signatures on the inputs for Redeem encrypted by A and B respectively.

This completes the scaffold. If the buyer wants to go through with the purchase they sign and broadcast the Fund transaction. Then, if the seller wants to go through with the sale they decrypt both one-time encrypted signatures on the Redeem's inputs with (a, b) and broadcast Redeem. From the one-timeness of the encryptions the buyer learns (a,b) and can therefore recover $r \leftarrow \alpha - a$ and $x \leftarrow \beta - b$.

References

[1] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.