Monero's Building Blocks
Part 8 of 10 – *Introduction to Pedersen Commitments and Confidential Transactions*

Bassam El Khoury Seguias

BTC: 3FcVvBZwTUkUrcqJd16RcjR42qT2tDWHWn

ETH: 0xb79Fb9194C8Cc6221368bb70976e18609Ab9AcA8

April 24, 2018

1 Introduction

Pedersen Commitments are at the heart of how Monero conceals transaction amounts. Confidential transactions as enabled by Pedersen Commitments were outlined and defined by Gregory Maxwell in [2]. In what follows we first introduce the notion of a group homomorphism (of which the Pedersen Commitment map is a particular instance), we then define the Pedersen Commitment map, and finally present the mechanisms of a confidential transaction enabled by a such a map.

2 Group homomorphism

Let $(M, \boxplus)$ and $(N, \oplus)$ be 2 groups with respective group operations $\boxplus$ and $\oplus$. A function $f : M \to N$ is called a group homomorphism if and only if $f(u \boxplus v) = f(u) \oplus f(v),\ \forall u, v \in M$. In other terms, operating on 2 elements in M and then applying f is equivalent to applying f on each element separately and then operating on the 2 outputs in N.

We now introduce a specific instance of a group homomorphism that we will invoke when concealing transaction amounts with Monero. In parts 5, 6 and 7, we defined a large finite group generated by a particular elliptic curve whose equation is given by:

$$E : -x^2 + y^2 = 1 + dx^2y^2$$

For a concise introduction to elliptic curve cryptography, one could consult the post entitled *Elliptic Curve Groups*. We recall that the above equation is a polynomial over $\mathbb{F}_q$ where q is a very large prime and d is a pre-defined element of $\mathbb{F}_q$. We refer to the group generated by this elliptic curve as $E(\mathbb{F}_q)$. We reiterate the following observations mentioned in previous parts:

- Elements of $E(\mathbb{F}_q)$ are pairs $(x, y) \in \mathbb{F}_q^2$ that satisfy the above equation.

- Elliptic curve groups in general and $E(\mathbb{F}_q)$ in particular have a well defined addition operation that we denote by $\oplus$.
- $E(\mathbb{F}_q)$ contains a special element G (not necessarily unique) that we refer to as the base point. The base point has order $l < q$, where l is a very large prime. That means that adding G to itself l times yields the identity element e of $E(\mathbb{F}_q)$. In other terms, $G \oplus ... \oplus G = e$. We simply write $l \otimes G = e$ (the notation $\otimes$ serves as a reminder that this is scalar multiplication associated with $\oplus$).
- We let $\{G\}$ denote the group generated by G under the $\oplus$ operation of $E(\mathbb{F}_q)$.

Let $(N, \oplus) \equiv (\{G\}, \oplus)$, and let $(M, \boxplus) \equiv (\mathbb{F}_l \times \mathbb{F}_l, +)$ where $+$ denotes element-wise addition in modulo l arithmetic over $\mathbb{F}_l \times \mathbb{F}_l$.

It is a known result in group theory that if a is a generator of a cyclic group $\{a\}$ of order m, then there are $\phi(m)$ elements of the group that have order m (ϕ is the euler function introduced in part 1). In our case, the generator G of $\{G\}$ has prime order l. Moreover $\phi(l) = l - 1$ (since l is prime). Hence we can find $l - 1$ other generators of $\{G\}$. Let $H \neq G$ be another generator such that the DL (discrete logarithm) of H with respect to G is unknown. We define the **Pedersen Commitment** map as follows:

$$k : \mathbb{F}_l \times \mathbb{F}_l \to \{G\}$$

$$(x, a) \to k(x, a) \equiv (x \otimes G) \oplus (a \otimes H)$$

We claim that the map k is additively homomorphic. To see why, let $(x_1, a_1),\ (x_2, a_2)\ \in \mathbb{F}_l \times \mathbb{F}_l$. We then have:

$$k(x_1, a_1) \oplus k(x_2, a_2) = [(x_1 \otimes G) \oplus (a_1 \otimes H)] \oplus [(x_2 \otimes G) \oplus (a_2 \otimes H)]$$

$$= ((x_1 + x_2) \otimes G)) \oplus ((a_1 + a_2) \otimes H)) \text{ (where } + \text{ denotes } \pmod{l} \text{ over } \mathbb{F}_l)$$

$$= k((x_1 + x_2), (a_1 + a_2)) \ = \ k((x_1, a_1) + (x_2, a_2)), \text{ hence } k \text{ is homomorphic.}$$

We call $k(x, a)$ a commitment. a denotes the amount we commit to, while x is referred to as the blinding factor. Note that $\forall c \in \{G\}, \forall a \in \mathbb{F}_l$, there always exists a blinding factor $x \in \mathbb{F}_l$ such that $k(x, a) = c$. Indeed, given c and a, an adequate x must satisfy $(x \otimes G) \oplus (a \otimes H) = c$ (by definition of the map k). This is equivalent to finding x such that $x \otimes G = c \ominus (a \otimes H)$. ($\ominus$ denotes the additive inverse of $\oplus$ over the group $\{G\}$). Since $c \ominus (a \otimes H) \in \{G\}$ and since G is a generator of $\{G\}$, we can be certain of the existence of such an x.

Note that this does not mean that we can find the value of x since this would require finding the DL of $c \ominus (a \otimes H)$ in base G. However, it means that for a given amount a, one could achieve any commitment value $c \in \{G\}$ by appropriately choosing $x \in \mathbb{F}_l$. A consequence of this is if we are given a and we randomly choose $x \in \mathbb{F}_l$, then c would look random over $\{G\}$. So given a transaction amount $a \in \mathbb{F}_l$, one can randomly generate a blinding factor $x \in \mathbb{F}_l$ and calculate $k(x, a) \equiv (x \otimes G) \oplus (a \otimes H)$.

3 Confidential transaction

In Bitcoin, transaction amounts are openly published to allow the network to verify that no value was created out of thin air or destroyed. The Bitcoin network checks that for each transaction, the total input amount of relevant UTXOs (denoted by $(a_{in})_i,\ i \in \{1, ..., m\}$) is equal to that of the output UTXOs (denoted by $(a_{out})_j,\ j \in \{1, ..., t\}$). It must be that

$$\sum_{i=1}^{m}(a_{in})_i = \sum_{j=1}^{t}(a_{out})_j$$

The question that a confidential transaction scheme must answer is whether the above equation can be verified without accessing the exact transaction values $(a_{in})_i$ and $(a_{out})_j$. We now describe a method that solves the question by using the homomorphic Pedersen Commitment previously introduced. Without loss of generality:

- $\forall i \in \{1, ..., m\}$, let $(C_{in})_i = ((x_{in})_i \otimes G) \oplus ((a_{in})_i \otimes H)$ be the Pedersen Commitment associated with amount $(a_{in})_i$ with blinding factor $(x_{in})_i$ randomly chosen in $\mathbb{F}_l$.
- Let $(a_{out})_t \equiv txfee$ be the miner's transaction fee and let $(C_{out})_t = (a_{out})_j \otimes H$ be the Pedersen Commitment associated with txfee. The blinding factor $(x_{out})_t$ is deliberatly chosen to be 0 (i.e., the identity element of $\mathbb{F}_l$).
- $\forall j \in \{1, ..., t-1\}$, let $(C_{out})_j = ((x_{out})_j \otimes G) \oplus ((a_{out})_j \otimes H)$ be the Pedersen Commitment associated with amount $(a_{out})_j$ with blinding factor $(x_{out})_j$ randomly chosen in $\mathbb{F}_l$. We additionaly require that $\sum_{i=1}^{m}(x_{in})_i = \sum_{j=1}^{t-1}(x_{out})_j \pmod{l}$ (the rationale will become clear in the next paragraph).

Suppose that:

$$\sum_{i=1}^{m}(C_{in})_i = \sum_{j=1}^{t}(C_{out})_j$$

This is equivalent to:

$$\sum_{i=1}^{m}(C_{in})_i = \sum_{j=1}^{t-1}(C_{out})_j \ \oplus \ (txfee \otimes H)$$

(by definition of txfee and $(c_{out})_t$)

$$\Longleftrightarrow$$

$$\sum_{i=1}^{m}[((x_{in})_i \otimes G) \oplus ((a_{in})_i \otimes H)] = \sum_{j=1}^{t-1}[((x_{out})_j \otimes G) \oplus ((a_{out})_j \otimes H)] \ \oplus \ (txfee \otimes H)$$

$$\Longleftrightarrow$$

$$\sum_{i=1}^{m} k((x_{in})_i, (a_{in})_i) = \sum_{j=1}^{t-1} k((x_{out})_j, (a_{out})_j) \ \oplus \ (txfee \otimes H)$$

$$\Longleftrightarrow$$

$$k(\sum_{i=1}^{m}(x_{in})_i, \sum_{i=1}^{m}(a_{in})_i) = k(\sum_{j=1}^{t-1}(x_{out})_j, \sum_{j=1}^{t-1}(a_{out})_j) \ \oplus \ (txfee \otimes H)$$

(by invoking the additive homomorphic property of the Pedersen Commitment map)

$$\iff$$

$$[(\sum_{i=1}^{m}(x_{in})_i) \otimes G] \ \oplus \ [(\sum_{i=1}^{m}(a_{in})_i) \otimes H]$$

$$= [(\sum_{j=1}^{t-1}(x_{out})_j) \otimes G] \ \oplus \ [\sum_{j=1}^{t-1}(a_{out})_j) \otimes H] \ \oplus \ [txfee \otimes H]$$

$$\iff$$

$$[\sum_{i=1}^{m}(x_{in})_i \ - \ \sum_{j=1}^{t-1}(x_{out})_j] \otimes G = [txfee \ + \ \sum_{j=1}^{t-1}(a_{out})_j) \ - \ \sum_{i=1}^{m}(a_{in})_i)] \otimes H$$

where $+$ and $-$ are addition and subtraction in modulo l arithmetic over $\mathbb{F}_l$. Note that by design, the left hand side is equal to 0 (because $\sum_{i=1}^{m}(x_{in})_i = \sum_{j=1}^{t-1}(x_{out})_j \pmod{l}$). We can thus conclude that if $\sum_{i=1}^{m}(x_{in})_i = \sum_{j=1}^{t-1}(x_{out})_j \pmod{l}$, then:

$$\sum_{i=1}^{m}(C_{in})_i = \sum_{j=1}^{t}(C_{out})_j$$

$$\iff \sum_{i=1}^{m}(a_{in})_i = \sum_{j=1}^{t}(a_{out})_j \pmod{l}$$

By ensuring that

1. $\sum_{i=1}^{m}(x_{in})_i = \sum_{j=1}^{t-1}(x_{out})_j \pmod{l}$, and
2. $\forall i \in \{1, ..., m\}, \ \forall j \in \{1, ...t\}$ the amounts $(a_{in})_i$ and $(a_{out})_j$ remain confined to a pre-defined range $[0, 2^r] \subset \mathbb{F}_l$. r is chosen in such a way that 2^r is significantly smaller than l. More specifically, suppose m_{max} and t_{max} respectively denote the maximum number of inputs and outputs that can be used in any given transaction. By letting $r < min(log_2(\frac{l}{m_{max}}), log_2(\frac{l}{t_{max}}))$, we are guaranteed that:

$$\sum_{i=1}^{m}(a_{in})_i \leq m_{max} \times max_{i=1}^{m}(a_{in})_i \leq m_{max} \times 2^r < l$$

$$\sum_{j=1}^{t}(a_{out})_j \leq t_{max} \times max_{j=1}^{t}(a_{out})_j \leq t_{max} \times 2^r < l$$

we get the following equivalence:

$$\sum_{i=1}^{m}(C_{in})_i \ \ominus \ \sum_{j=1}^{t}(C_{out})_j = 0$$

$$\iff \sum_{i=1}^{m}(a_{in})_i - \sum_{j=1}^{t}(a_{out})_j = 0$$

It is important to note that the amounts balance out in actuality and not in the more relaxed $\pmod{l}$ sense. This is because of the constraint we imposed on all transaction amounts to be confined to the $[0, 2^r]$ range. If this constraint was no imposed, one would be able to create or destroy Monero currency while still maintaining a balanced equation. To see this, suppose transaction amounts can take on any value in $\mathbb{F}_l$ instead of being restricted to $[0, 2^r]$. Let $m = 3$ with $(a_{in})_1 = l - 4, \ (a_{in})_2 = 3, \ and (a_{in})_3 = 5$. Also let $t = 2$ with $(a_{out})_1 = 3, \ and (a_{out})_2 = 1$.

$$\text{Clearly, } \sum_{i=1}^{m}(a_{in})_i - \sum_{j=1}^{t}(a_{out})_j = l \neq 0.$$

$$\text{However, } \sum_{i=1}^{m}(a_{in})_i - \sum_{j=1}^{t}(a_{out})_j = 0 \pmod{l}.$$

If this transaction gets approved by the network, we would have effectively destroyed l units of currency. Conversely, exchanging the input and output values would allow the creation of l units of currency out of thin air. This example demonstrates the importance of having a balanced equation independent of modulo p arithmetic. By confining all transaction amounts to the $[0, 2^r]$ range, we ensure that this is the case. To prove that a transaction amount lies in a certain range, Monero makes use of the **Borromean signature** construct. We are not covering its mechanics in this work but the interested reader can consult [1].

The result above allows one to safely replace the transaction amounts by their respective Pedersen Commitments (i.e., hide the transaction amounts) while still ensuring proper accounting.

References

[1] G. Maxwell and A. Poelstra. Borromean ring signatures. -, 2015.

[2] Greg Maxwell. Confidential transactions, 2015.