Hidden amounts scheme on the base of a ring signature for independent generators

Anton A. Sokolov (acmxddk@gmail.com, anton@zano.org)
Zano

Preliminaries

Hidden amount

$A=fH_1+vH_2$, where

- A - hidden amount
- H_1, H_2 - independent generators, i.e. H_1!~H_2
- f - random uniform factor
- v - amount

Two hidden amounts are called v-equal iff their v's are equal. Thus, the A's equality implies v-equality, the converse is not always true.

Hashes

- **Hp**(X) - ideal point hash, takes a point X, returns point **Hp**(P)
- **Hs**(args) - ideal scalar hash, takes a list args of scalars and points, returns scalar **Hs**(args), sensitive to the order of entries in the args.

Blockchain, addresses and amounts

Blockchain is assumed to be a standard CryptoNote blockchain with the stealth addresses generated using the standard CN formula. The only three differences to the blockchain compared to the standard CN are

- For each stealth address P a hidden amount A is substituted for the associated with the P publicly seen amount v. Thus, v's don't exist in the blockchain any more, and A's are put instead of v's at their places.
- The blockchain doesn't use the standard CN ring signature any more, it uses a threshold version of ring signature with the properties described in the next section.
- Key image **Hp**(P)/x is substituted for the standard CN key image x**Hp**(P) everywhere (recalling P=xG, where x is a private key for stealth address P, and G is an independent generator).

Elementary proofs (used as building blocks)

Schnorr signature

Given two points: X and Y, the Schnorr signature provides a zk-proof for that the signer knows a scalar y such that Y=yX. The Schnorr signature size is the size of two scalars.

Generalized Schnorr signature (for 2 base generators)

Given two points: G_0, G_1 such the G_0!~G_1, and a point X, a generalized Schnorr signature provides a zk-proof for that the signer knows a pair of scalars (x_0, x_1) such that $X=x_0G_0+x_1G_1$.

We have a generalized Schnorr signature that has the size of three scalars.

Vector Schnorr signature

Given two point vectors: $[X_i]_{i=0...(K-1)}$ and $[Y_i]_{i=0...(K-1)}$, where K>0, a vector Schnorr signature provides a zk-proof for that the signer knows a scalar y such that $\forall i \in [0,K-1]$: $Y_i=yX_i$.
We have a vector Schnorr signature that for any K has the size of K+1 scalars.

Batch Schnorr signature

Given a point vector: $[X_i]_{i=0...(K-1)}$, where K>0, and a point G, a batch Schnorr signature provides a zk-proof for that the signer knows a scalar vector $[x_i]_{i=0...(K-1)}$ such that $\forall i \in [0,K-1]$: $X_i=x_iG$.
We have a batch Schnorr signature that for any K has a constant size, namely, the size of only two scalars.

Generalized batch Schnorr signature (for 2 base generators)

Given two points: G_0, G_1 such the G_0!~G_1, and a point vector $[X_i]_{i=0...(K-1)}$, where K>0, a generalized batch Schnorr signature provides a zk-proof for that the signer knows a vector of scalar pairs $[(x_{0i}, x_{1i})]_{i=0...(K-1)}$ such that $[X_i]_{i=0...(K-1)}=[x_{0i}G_0+x_{1i}G_1]_{i=0...(K-1)}$.
We have a generalized batch Schnorr signature that has the size of three scalars.

Threshold L out of N ring signature

Given a ring of N points $[X_i]_{i=0...(N-1)}$ such that all X's in the $[X_i]_{i=0...(N-1)}$ are independent (in the sense of the discrete logarithm relationship) of each other and of G, H_0, H_1, H_2, where G is a generator used for the stealth addresses, H_1, H_2 are generators used for the hidden amounts, H_0 is an one more independent generator reserved for the later use, and given a list of L points $[G_j]_{j=0...(L-1)}$, a threshold ring signature provides a zk-proof for that the signer knows L relations $[G_j\sim S_j]_{j=0...(L-1)}$ such that $[S_j]_{j=0...(L-1)} \subset [X_i]_{i=0...(N-1)}$. Note, the signature itself doesn't prove that $\forall i,j$: $i\neq j \Rightarrow S_i\neq S_j$, this is expected to be proven by the other means.
We have a threshold L out of N signature that has a logarithmic in N size.

Random weighting (for one weight)

Given two pairs of points (A, B) and (X, Y) such that A!~B, given a random weight z=**Hs**(A, B, X, Y, optional args ...), a proof of (A+zB)~(X+zY) implies a proof of (A, B)~(X, Y).

Random weighting (for two weights)

Given two 3-tuples of points (A, B, C) and (X, Y, Z) such that C!=lin(A, B) and X!~Y, given the random weights z_0=**Hs**(A, B, C, X, Y, Z, optional args ...) and z_1=**Hs**(z_0), a proof of $(A+z_0B+z_1C)\sim(X+z_0Y+z_1Z)$ implies a proof of (A, B, C)~(X, Y, Z).

Linearity (over two generators)

Given four points A, B, X, Y and two independent generators H_0, H_1 (such that H_0!~H_1), given the proofs of A~H_0, X~H_0, B~H_1, Y~H_1, a proof of (A+B)~(X+Y) implies a proof of (A, B)~(X, Y).

Linearity (over three generators)

Given six points A, B, C, X, Y, Z and three independent generators H_0, H_1, H_2 (such that ort(H_0, H_1, H_2)), given the proofs of A~H_0, X~H_0, B~H_1, Y~H_1, C~H_2, Z~H_2, a proof of (A+B+C)~(X+Y+Z) implies a proof of (A, B, C)~(X, Y, Z).

Scheme

The scheme generates a zk-proof for the following five facts:

I. Signer knows L private keys for L distinct public keys (stealth addresses) in a ring of N public keys $[P_i]_{i=0...(N-1)}$.
II. The key images for those L public keys, that the signer knows private keys for, are $[I_j]_{j=0...(L-1)}$.
III. Signer knows openings $[(f'_j, v'_j)]_{j=0...(L-1)}$ for L hidden amounts $[A'_j]_{j=0...(L-1)}$ corresponding to those L public keys, that the signer knows private keys for.
IV. Signer knows openings $[(g_j, e_j)]_{j=0...(M-1)}$ for M output hidden amounts $[E_j]_{j=0...(M-1)}$.
V. The sum of all hidden amounts $[A'_j]_{j=0...(L-1)}$ is v-equal to the sum of all hidden amounts $[E_j]_{j=0...(M-1)}$ corresponding to the M outputs.

Limitation: the scheme doesn't check the hidden amounts against the range overflow. A separate range proof, e.g. the Bulletproofs algorithm, is to be applied to the output hidden amounts $[E_j]_{j=0...(M-1)}$ to prevent the range overflow.
Note: zk is meant in that sense that no information beyond the mentioned five facts is revealed.

Publicly known fixed generators

There are four known fixed generators H_0, H_1, H_2, G such that ort(H_0, H_1, H_2, G) holds and, moreover, it's guaranteed that the unknown scalars (h_0, h_1, h_2) such that (H_0, H_1, H_2)=(h_0G, h_1G, h_2G) are distributed uniformly at random. This can be achieved, for instance, by defining (H_0, H_1, H_2, G)=(**Hp**(3G), **Hp**(2G), **Hp**(G), G).

Publicly seen mixins, outputs, hidden amounts, and key images

- N - number of mixins
- L - number of secret inputs
- M - number of outputs
- $[(P_i, A_i)]_{i=0...(N-1)}$ - ring of mixins, i.e. ring of (stealth address, hidden amount) pairs
- $[(R_i, E_i)]_{i=0...(M-1)}$ - list of outputs, i.e. list of (stealth address, hidden amount) output pairs
- $[I_i]_{i=0...(L-1)}$ - list of key images.

Note: L inputs are not known (neither directly nor indirectly), as they are anonymously picked from N mixins.

Signer's private data

- Signer knows L scalar 3-tuples $[(x'_j, f'_j, v'_j)]_{j=0...(L-1)}$ such that L pairs $[(P'_j, A'_j)]_{j=0...(L-1)}=[(x'_jG, f'_jH_1+v'_jH_2)]_{j=0...(L-1)}$ correspond to some L distinct pairs in the ring $[(P_i, A_i)]_{i=0...(N-1)}$.
- Signer knows M scalar pairs $[(g_j, e_j)]_{j=0...(M-1)}$ such that $[E_j]_{j=0...(M-1)}=[g_jH_1+e_jH_2]_{j=0...(M-1)}$.

Signer's algorithm

1. Generate L random scalars $[r_j]_{j=0...(L-1)}$ and build L 4-tuples $[(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}=[(r_jH_0, r_jA'_j, r_jP'_j, r_j\mathbf{Hp}(P'_j))]_{j=0...(L-1)}$. Publicate $[(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}$.
2. Build two random weights (z_0, z_1)
   - $z_0=\mathbf{Hs}([(P_i, A_i)]_{i=0...(N-1)}, [(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}, [I_i]_{i=0...(L-1)})$
   - $z_1=\mathbf{Hs}(z_0)$
3. Build a ring of N points $[X_i]_{i=0...(N-1)}=[H_0+A_i+z_0P_i+z_1\mathbf{Hp}(P_i)]_{i=0...(N-1)}$
4. Build L points $[G_j]_{j=0...(L-1)}=[T_j+B_j+z_0U_j+z_1Y_j]_{j=0...(L-1)}$
5. Using the threshold L out of N ring signature, build a proof of $[G_j\sim S_j]_{j=0...(L-1)}$, where $[S_j]_{j=0...(L-1)} \subset [X_i]_{i=0...(N-1)}$. Publicate this proof.
6. Using L vector Schnorr signatures, build a proof of $[(U_j, Y_j)\sim(G, I_j)]_{j=0...(L-1)}$. Publicate this proof.
7. Generate L random scalars $[k_j]_{j=0...(L-1)}$ and build L points $[K_j]_{j=0...(L-1)}=[k_jH_1]_{j=0...(L-1)}$. Publicate $[K_j]_{j=0...(L-1)}$.
8. Using the batch Schnorr signature, build a proof of $[(K_j\sim H_1)]_{j=0...(L-1)}$. Publicate this proof.
9. Knowing L scalars $[1/r_j]_{j=0...(L-1)}$, build L points $[W_j]_{j=0...(L-1)}=[(B_j+K_j)/r_j]_{j=0...(L-1)}$. Publicate $[W_j]_{j=0...(L-1)}$.
10. Using L vector Schnorr signatures, build a proof of $[(T_j, B_j+K_j)\sim(H_0, W_j)]_{j=0...(L-1)}$. Publicate this proof.

11. Using the generalized batch Schnorr signature, build a proof of that the following L+M relations are known to the signer: $[W_j=\text{lin}(H_1,H_2)]_{j=0...(L-1)}$ and $[E_j=\text{lin}(H_1,H_2)]_{j=0...(M-1)}$. Publicate this proof.
12. Let $D=\sum_{j=0...(L-1)}W_j-\sum_{j=0...(M-1)}E_j$. Using the Schnorr signature, build a proof of $D\sim H_1$. Publicate this proof.

Preprocessing steps

Prior to running the signing algorithm the signer performs the following steps

- Select the ring of N mixins $[(P_i, A_i)]_{i=0...(N-1)}$
- Assign values to the key images $[I_j]_{j=0...(L-1)}=[Hp(P'_j)/x'_j)]_{j=0...(L-1)}$, where all L P'_js are different addresses from the ring. Thus, all L key images are different from each other.
- Generate M output addresses $[R_i]_{i=0...(M-1)}$ using the standard CN formula
- Assign values to the output amounts $[e_i]_{i=0...(M-1)}$ holding the equality $\sum_{j=0...(L-1)}v'_j=\sum_{j=0...(M-1)}e_j$
- Generate M output amount random factors $[g_i]_{i=0...(M-1)}$
- Let $[(R_i, E_i)]_{i=0...(M-1)}=[(R_i, g_jH_1+e_jH_2)]_{j=0...(M-1)}$

Postprocessing steps

After performing the signing algorithm the signer

- Encrypts each pair in the list of scalar pairs $[(g_j, e_j)]_{j=0...(M-1)}$ with the corresponding output address from the list of the output addresses $[R_i]_{i=0...(M-1)}$ and publishes the encrypted pairs.
- Sends all the published data and proofs to the verifiers and to the receiver.

Transmitted (additionally publicly seen) data

• $[(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}$	= 4L points
• Threshold L out of N ring signature for $[G_j\sim S_j]_{j=0...(L-1)}$	=~ log(N), =~ L
• L vector Schnorr signatures for $[(U_j, Y_j)\sim(G, I_j)]_{j=0...(L-1)}$	= 3L scalars
• $[K_j]_{j=0...(L-1)}$	= L points
• Batch Schnorr signature for $[K_j\sim H_1]_{j=0...(L-1)}$	= 2 scalars
• $[W_j]_{j=0...(L-1)}$	= L points
• L vector Schnorr signatures for $[(T_j, B_j+K_j)\sim(H_0, W_j)]_{j=0...(L-1)}$	= 3L scalars
• Generalized batch Schnorr signature for the $[W_j=\text{lin}(H_1,H_2)]_{j=0...(L-1)}$ and $[E_j=\text{lin}(H_1,H_2)]_{j=0...(M-1)}$	= 3 scalars
• Schnorr signature for $D\sim H_1$	= 2 scalars

Verifier's algorithm

1. Restore the two random weights (z_0, z_1)
   - $z_0=\mathbf{Hs}([(P_i, A_i)]_{i=0...(N-1)}, [(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}, [I_i]_{i=0...(L-1)})$
   - $z_1=\mathbf{Hs}(z_0)$
2. Restore the ring of N points $[X_i]_{i=0...(N-1)}=[H_0+A_i+z_0P_i+z_1\mathbf{Hp}(P_i)]_{i=0...(N-1)}$
3. Restore L points $[G_j]_{j=0...(L-1)}=[T_j+B_j+z_0U_j+z_1Y_j]_{j=0...(L-1)}$
4. Check the threshold L out of N ring signature for $[G_j\sim S_j]_{j=0...(L-1)}$, where $[S_j]_{j=0...(L-1)} \subset [X_i]_{i=0...(N-1)}$. Note, the statement $\forall i,j: i\neq j \Rightarrow S_i\neq S_j$ is not checked at this point
5. Check L vector Schnorr signatures for $[(U_j, Y_j)\sim(G, I_j)]_{j=0...(L-1)}$
6. Check the batch Schnorr signature for $[K_j\sim H_1]_{j=0...(L-1)}$
7. Check L vector Schnorr signatures for $[(T_j, B_j+K_j)\sim(H_0, W_j)]_{j=0...(L-1)}$
8. Check the generalized batch Schnorr signature for $[W_j=\text{lin}(H_1, H_2)]_{j=0...(L-1)}$ and $[E_j=\text{lin}(H_1, H_2)]_{j=0...(M-1)}$
9. Restore $D=\sum_{j=0...(L-1)}W_j-\sum_{j=0...(M-1)}E_j$
10. Check the Schnorr signature for $D\sim H_1$

How the verifier gets convinced in the (mentioned above) five facts

Preliminary information about the hidden amounts and addresses in the ring

- From the previously generated proofs in the blockchain the verifier is convinced that each hidden amount in the ring is composed of H_1 and H_2, i.e., $\forall i \in [0, N\text{-}1]$: $A_i = \text{lin}(H_1, H_2)$.
- No information about the ring addresses is provided beforehand, e.g., there is no supposition that $\exists i \in [0, N\text{-}1]$: $P_i \sim G$, although later, according to the proof provided by the signer, it appears that such relations are known to the signer for L addresses in the ring.

(Section*) Verifier gets convinced in $[(T_j, B_j, U_j, Y_j) \sim (H_0, A'_j, P'_j, \mathbf{Hp}(P'_j))]_{j=0\ldots(L-1)}$

1. From the check 4 the verifier is convinced in L relations $[(T_j+B_j+z_0U_j+z_1Y_j) \sim (H_0+A'_j+z_0P'_j+z_1\mathbf{Hp}(P'_j))]_{j=0\ldots(L-1)}$, where $[(P'_j, A'_j)]_{j=0\ldots(L-1)}$ is a subset of possibly duplicated pairs such that $[(P'_j, A'_j)]_{j=0\ldots(L-1)} \subset [(P_i, A_i)]_{i=0\ldots(N-1)}$.
2. From the preliminary information about the hidden amounts the verifier is convinced that $\forall j$: $A'_j = \text{lin}(H_1, H_2)$. Hence, by definition of **Hp** the verifier is convinced that $\forall j$: $\mathbf{Hp}(P'_j) != \text{lin}(H_0+A'_j, P'_j)$.
3. From the check 5 the verifier is convinced that $\forall j$: $U_j \sim G$.
4. From the check 7 the verifier is convinced that $\forall j$: $T_j \sim H_0$.
5. From the checks 8 and 7 the verifier is convinced that $\forall j$: $W_j = \text{lin}(H_1, H_2)$ and $(B_j+K_j) \sim W_j$. Hence, it is convinced that $\forall j$: $(B_j+K_j) = \text{lin}(H_1, H_2)$.
6. From the check 6 the verifier is convinced that $\forall j$: $K_j \sim H_1$. Hence, it is convinced that $\forall j$: $B_j = \text{lin}(H_1, H_2)$. Also, as it is already convinced that $\forall j$: $T_j \sim H_0$, it is convinced that $\forall j$: $(T_j+B_j) = \text{lin}(H_0, H_1, H_2)$.
7. As $\forall j$: $(T_j+B_j) = \text{lin}(H_0, H_1, H_2)$ and $U_j \sim G$, the verifier is convinced that $\forall j$: $(T_j+B_j) !\sim U_j$.
8. As $\forall j$: $(T_j+B_j+z_0U_j+z_1Y_j) \sim (H_0+A'_j+z_0P'_j+z_1\mathbf{Hp}(P'_j))$, where $\mathbf{Hp}(P'_j) != \text{lin}(H_0+A'_j, P'_j)$ and $(T_j+B_j) !\sim U_j$, by the random weighting the verifier is convinced that $\forall j$: $(T_j+B_j, U_j, Y_j) \sim (H_0+A'_j, P'_j, \mathbf{Hp}(P'_j))$. Thus, from the linearity of the T_j+B_j and $H_0+A'_j$ it is convinced that $\forall j$: $(T_j, B_j, U_j, Y_j) \sim (H_0, A'_j, P'_j, \mathbf{Hp}(P'_j))$.

Facts I and II

1. From the Section*.8 and from the check 5 the verifier is convinced that $\forall j$: $(T_j, B_j, U_j, Y_j) \sim (H_0, A'_j, P'_j, \mathbf{Hp}(P'_j))$ and $(U_j, Y_j) \sim (G, I_j)$. Hence, it is convinced that $\forall j$: $(P'_j, \mathbf{Hp}(P'_j)) \sim (G, I_j)$. As all L I_js are different, the verifier is convinced that all L P'_js are different addresses from the ring. Hence, it is convinced that the subset $[(P'_j, A'_j)]_{j=0\ldots(L-1)}$ contains no duplicates, i.e., that all the L 4-tuples (T_j, B_j, U_j, Y_j) correspond to the different entries of the ring.
2. Thus, the verifier is convinced that $\forall j$: the signer knows a scalar x_j such that $P'_j = x_jG$ and $I_j = \mathbf{Hp}(P'_j)/x_j$.

Fact III

1. From the Section*.8 and Section*.6 the verifier is convinced that $\forall j$: $(T_j, B_j, U_j, Y_j) \sim (H_0, A'_j, P'_j, \mathbf{Hp}(P'_j))$ and that $B_j = \text{lin}(H_1, H_2)$. Hence, it is convinced that $\forall j$: $A'_j = \text{lin}(H_1, H_2)$, where all A'_js correspond to the P'_js from the ring.
2. Thus, the verifier is convinced that $\forall j$: the signer knows an opening for A'_j.

Fact IV

1. From the check 8 the verifier is convinced that $\forall j \in [0, M\text{-}1]$: $E_j = \text{lin}(H_1, H_2)$.
2. Thus, the verifier is convinced that the signer knows openings for all the output hidden amounts $[E_j]_{j=0\ldots(M-1)}$.

Fact V

1. From the checks 8, 10 the verifier is convinced that
   a. $\forall j \in [0, L\text{-}1]$: W_j is composed of H_1, H_2,
   b. $\forall j \in [0, M\text{-}1]$: E_j is composed of H_1, H_2,
   c. The sums $\sum_{j=0\ldots(L-1)} W_j$ and $\sum_{j=0\ldots(M-1)} E_j$ are v-equal to each other.
2. From the check 7 the verifier is convinced that $\forall j \in [0, L\text{-}1]$: there is a known to the signer scalar y_j such that $W_j = y_j(B_j+K_j)$ and, at the same time, $H_0 = y_jT_j$.

3. From the Section*.8 the verifier is convinced that $\forall j \in [0, L-1]$: there is a known to the signer scalar r_j such that $B_j=r_jA'_j$ and, at the same time, $T_j=r_jH_0$. Thus, the verifier is convinced that $\forall j \in [0, L-1]$: $y_j=1/r_j$.
4. From the check 6 the verifier is convinced that $\forall j \in [0, L-1]$: there is a known to the signer scalar k_j such that $K_j=k_jH_1$. Hence, the verifier is convinced that $\forall j \in [0, L-1]$: there are known to the signer scalars r_j, k_j such that $W_j=A'_j+(k_j/r_j)H_1$.
5. Hence, the verifier is convinced that the H_2 part of the sum $\sum_{j=0...(L-1)}W_j$ is equal to H_2 part of the sum $\sum_{j=0...(L-1)}A'_j$. Thus, the verifier is convinced that the sum $\sum_{j=0...(L-1)}A'_j$ of all the hidden amounts corresponding to those ring addresses that the signer has proven knowledge of the private keys for and which have the key images $[I_j]_{j=0...(L-1)}$ is v-equal to the sum $\sum_{j=0...(M-1)}E_j$ of all the output hidden amounts.

No information is revealed beyond the (above) five facts

The following additional points are publicly seen compared to the standard CN, as all they are indistinguishable from points generated from G by multiplying it by distinct private (uniformly) random scalars:

- $[(T_j, B_j, U_j, Y_j)]_{j=0...(L-1)}$ - $\forall j$: ort(T_j, B_j, U_j, Y_j) and $\exists$ random r_j such that $(T_j, B_j, U_j, Y_j)=(r_jH_0, r_jA'_j, r_jP'_j, r_j\mathbf{Hp}(P'_j))$
- $[K_j]_{j=0...(L-1)}$ - $\forall j$ $\exists$ random k_j: $K_j=k_jH_1$
- $[E_j]_{j=0...(M-1)}$ - $\forall j$ $\exists$ random factor g_j: $E_j=g_jH_1+e_jH_2$
- $[W_j]_{j=0...(L-1)}$ - $\forall j$: $W_j=A'_j+(k_j/r_j)H_1$

Here is the sketch of a proof for W_j's: as the sender (it is assumed adversarial) knows opening for A'_j, the problem reduces to the question if $(k_j/r_j)H_1$ is indistinguishable from cH_1, where c is some uniformly random scalar. We have the value of r_jH_1 exposed due to the sender's knowledge of A'_js opening, also we have the k_jH_1 exposed. Thus, we have a DDDH 4-tuple: $(H_1, k_jH_1, r_jH_1, (k_j/r_j)H_1)$ that is indistinguishable from the independent randomness according to the DDDH assumption, that holds together with the DDH assumption (DDDH<=>DDH).

Receiver's algorithm

- Run the verifier's algorithm
- Decrypt the hidden amount opening (g, e)
- Optionally, send the received amount to anyone else using the sender's algorithm