POLITECNICO
MILANO 1863

Addressing Privacy and Fungibility Issues in Bitcoin: Confidential Transactions

Alessandro Miola

Supervisors: Daniele Marazzina,
Ferdinando M. Ametrano

20th December 2018

Introduction

- Privacy is fundamental in every financial and monetary system. Bitcoin should not make any exception.
- Bitcoin is neither confidential nor anonymous, but rather pseudonymous.
- Both Bitcoin's blockchain structure and security model seem not to be ideal for privacy.
- Lack of privacy even affects Bitcoin's capacity to serve as money $\Rightarrow$ detrimental for fungibility: not all bitcoins are equal.

Outline

Bitcoin's transactions

- bitcoins exist as *unspent transaction outputs* (UTXO).
- Transaction outputs are indivisible chunks of currency recorded on the blockchain and associated to addresses:
    - they embed the amount and the mathematical puzzle (*locking script*) which determines the conditions for spending.
- List of inputs referencing and spending UTXO and generating new ones:
    - they holds a pointer to the consumed UTXO and the *unlocking script* satisfying the conditions for spending;
    - the unlocking script generally must hold a digital signature proving ownership of the referenced output.

Outline

Privacy

- Transactional graph privacy: who is paying who?
  - Linkability of transactions;
  - bad users' practices (*address reuse*).
- Value privacy: how much is one paying or receiving?
  - Unencrypted transaction amounts.
- Identity privacy: who is behind the coins?
  - Leakage of personal information when accessing exchanges or on-line stores.
- Trade-off: blockchain's transparency $\Leftrightarrow$ Bitcoin's privacy.

Fungibility

- Property of a unit of a good to be indistinguishable from any other unit of the same good.
- Fundamental property for moneys and currencies:
  - do not want to care of the possibility of possession of banknotes being revoked for their “bad” history;
  - possibility of *blacklisting* banknotes would destroy confidence in receiving them.
- Bitcoin is substantially *immediate* and *final* payment; this makes discussion intertwine with money in its cash-like forms.
- Trade-off: blockchain’s transparency $\Leftrightarrow$ bitcoin’s fungibility.

Outline

Pedersen commitment

- *Commitment scheme*: keep a piece of data secret, but commit to it to prevent tampering.

Definition.
Let $(\mathbb{G}, \circ)$ be an elliptic curve group of prime order n. Let G, H be two NUMS generators of $\mathbb{G}$; let r, $v \in \mathbb{Z}_n$ (r chosen at random). Define a Pedersen commitment C to v by the following scheme:

$$\begin{aligned} \mathit{commit}\colon \mathbb{Z}_n^2 &\to \mathbb{G} \\ (r, v) &\mapsto rG + vH \\ \mathit{open}\colon \mathbb{Z}_n^2 \times \mathbb{G} &\to \{\mathit{True}, \mathit{False}\} \end{aligned}$$

such that $\mathit{open}(r, v, C) \mapsto \mathit{True}\ \forall (r, v)$ in the domain of *commit*.

Pedersen commitment: properties

- *Perfectly hiding* and *computationally binding* commitment scheme.

Definition.

A commitment scheme is said to be:

- perfectly (computationally) *hiding* if the distribution of $commit(r, v)$ for uniformly random r is equal (computationally indistinguishable) for fixed values of v;
- perfectly *binding* if $\forall(r, v)$ in the domain of $commit$, $\nexists(r', v') \neq (r, v)$: $open(r', v', commit(r, v)) \mapsto True$; computationally *binding* if no PPT (probabilistic polynomial time) algorithm can produce such (r', v') with non-negligible probability.

Pedersen commitment: properties

- *Perfectly hiding* and *computationally binding* commitment scheme.
- *Additively homomorphic* commitment scheme.

Definition.
A commitment scheme is *additively homomorphic* if:

- $commit(r, v) + commit(r', v') = commit(r + r', v + v')$.

Zero-Knowledge proof of knowledge

- Proof that yields nothing but its validity.
    - Alice tries to convince Bob of being in possession of some secret information.
    - Proof performed in Zero-Knowledge.

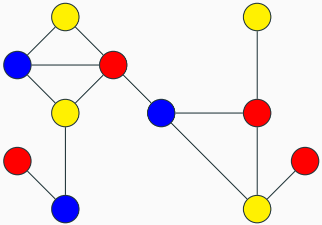

Figure 1: Graph 3-colorability problem

Ring signature

- OR proof: given a ring of r public keys $\{Q_0, \ldots, Q_{r-1}\}$, the ambiguous signer proves to know $\{q_0$ OR q_1 OR $\ldots$ OR $q_{r-1}\}$.

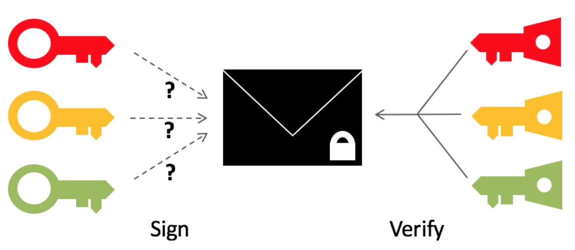

Figure 2: Ring signature scheme

- Tool for whistleblowing.

Ring signature

AOS_SIGN(m, q_{i^*}, Q_i: $0 \le i \le r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^*+1, \ldots, r-1, 0, \ldots, i^*-1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
    2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
    3. $K_i \leftarrow s_i G - e_i Q_i$;
4. $e_{i^*} \leftarrow hash(K_{i^*-1} || m || i^*)$;
5. $s_{i^*} \leftarrow k_{i^*} + e_{i^*} q_{i^*}$;
6. **return** $(e_0, s_0, \ldots, s_{r-1}) =: \sigma$

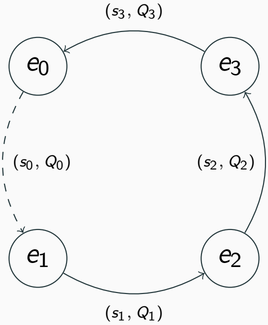

Figure 3: AOS ring signature (1-of-4)

Adapted from: [5]

Ring signature

AOS_VERIFY(m, σ, Q_i: $0 \le i \le r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i || m || i + 1\%r)$;
2. **if** $e_0 = 0$ or $e_0 \ge n$:
   return False;
3. **if** $e_0 = \sigma[0]$:
   return True;

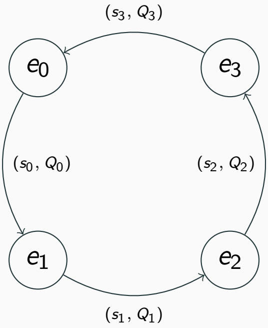

Figure 3: AOS ring signature verification (1-of-4)

Adapted from: [5]

Outline

Confidential Transactions

- Proposal for a transactional format with encrypted amounts, which requires the same cryptographic assumptions of Bitcoin (hardness of ECDLP).
- Built through homomorphic encryption without compromising the possibility for the nodes to verify the validity of each transaction.
- It provides *value privacy*.

Pedersen commitment in CT

- Substitute 8-byte output amounts in the clear with *perfectly hiding* 33-byte Pedersen commitments to the amounts.
- Interpretation of the parameters:
  - r: secret random *blinding factor*;
  - v: committed *output amount*.

$$v_{i1} = v_{o1} + v_{o2} + fee$$

$$\Downarrow$$

$$(r_{i1}G + v_{i1}H) = (r_{o1}G + v_{o1}H) + (r_{o2}G + v_{o2}H) + fH$$

$$\Downarrow$$

$$r_{i1} = r_{o1} + r_{o2}$$

$$v_{i1} = v_{o1} + v_{o2} + f$$

Commitment to value 0

- Instrumental in verifying the validity of each transaction.
- A commitment $C = rG$ gives the opportunity to produce a digital signature with the commitment as public key:
  - by definition, a signature with private key q can be verified with public key qG;
  - if $v \neq 0$, it is infeasible to find q such that $qG = rG + vH$.

$$\Downarrow$$

A Pedersen commitment can be proven to be a commitment to $v = 0$ by signing the transaction with r as private key, C as public key.

Zero-Knowledge range proof

Reason for:

- addition is modular and wraps around;
- possible to create money from nothing.

Example.

Consider a curve built on a finite field of prime order $n = 13$.

Inputs	Outputs
$C(r_{i1}, 2)$	$C(r_{o1}, 8)$
	$C(r_{o2}, 7)$

Table 1: Example of wrapping

Inputs	Outputs
$C(r_{i1}, 2)$	$C(r_{o1}, 5)$
	$C(r_{o2}, -3)$

Table 2: Negative amounts

Additional piece of data proving that each committed output is in a given range ensuring that no overflow is possible and the amount is non-negative.

Enforce Zero-Knowledgeness: Borromean ring signature

- Given r rings of public keys, the ambiguous signer proves to know one of $\{q_{0,0}$ OR $q_{0,1}$ OR $\ldots\}$ AND one of $\{q_{1,0}$ OR $q_{1,1}$ OR $\ldots\}$ AND $\ldots$ AND one of $\{q_{r-1,0}$ OR $q_{r-1,1}$ OR $\ldots\}$.

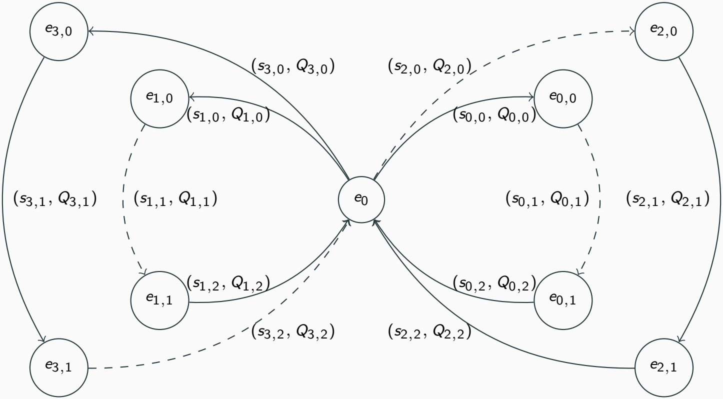

Figure 4: Borromean ring signatures

Enforce Zero-Knowledgeness: Borromean ring signature

- Given r rings of public keys, the ambiguous signer proves to know one of $\{q_{0,0}$ OR $q_{0,1}$ OR $\dots\}$ AND one of $\{q_{1,0}$ OR $q_{1,1}$ OR $\dots\}$ AND $\dots$ AND one of $\{q_{r-1,0}$ OR $q_{r-1,1}$ OR $\dots\}$.

	Signature size
r AOS ring signatures	$r \cdot N + r$ (32-bytes numbers)
Borromean ring signature	$r \cdot N + 1$ (32-bytes numbers)
Δ	**r - 1** (32-bytes numbers)

Table 3: Borromean ring signature: signature size

Use of Borromean ring signatures

- Consider the output amount in its base-4 expansion:

$$v = v_0 \cdot 4^0 + v_1 \cdot 4^1 + v_2 \cdot 4^2 + \cdots + v_{15} \cdot 4^{15};$$

Ring-sign over each digit:

- $C_i = r_i G + v_i 4^i H$, $i = 0, \ldots, 15$;
- arrange a ring of signatures per digit with a verification public key per digit value v_i, $i = 0, \ldots, 3$;
- provide a Borromean ring signature over the rings:

$$\{r_i G + v_i 4^i H, r_i G + v_i 4^i H - 4^i H, r_i G + v_i 4^i H - 2 \cdot 4^i H, r_i G + v_i 4^i H - 3 \cdot 4^i H\}$$

- $RP_v = (C_0, \ldots, C_{15}, \underbrace{e_0, s_{0,0}, \ldots, s_{0,3}, \ldots, s_{15,0}, \ldots, s_{15,3}}_{\textit{signature}})$.

Outline

Conclusions

- Confidential transactions would provide consistent value privacy in the protocol.
- Not ready yet for integration in the Bitcoin protocol as they suffer from excessively burdening each transaction size.

However,

- a new and more efficient solution to range proof construction has been proposed, *Bulletproofs*:
  - aggregation of range proofs;
  - batched verification of multiple proofs.
- *Mimblewimble*: promising cryptosystem built on confidential transactions.

References i

[1] M. Abe, M. Ohkubo, and K. Suzuki. **1-out-of-n signatures from a variety of keys.** https://www.iacr.org/archive/asiacrypt2002/25010414/25010414.ps, 2002.

[2] A. Gibson. **An investigation into confidential transactions.** https://github.com/AdamISZ/ConfidentialTransactionsDoc/blob/master/essayonCT.pdf.

[3] G. Maxwell. **Confidential transactions.** https://people.xiph.org/~greg/confidential_values.txt, 2015.

References ii

[4] G. Maxwell. **Improve transaction privacy / fungibility in bitcoin core and the bitcoin system [meta tracking issues].** https://github.com/bitcoin/bitcoin/issues/6568, 2015.

[5] G. Maxwell and A. Poelstra. **Borromean ring signatures.** https://github.com/ElementsProject/borromean-signatures-writeup.

[6] S. Nakamoto. **Bitcoin: A peer-to-peer electronic cash system.** http://bitcoin.org/bitcoin.pdf, 2008.

[7] T. P. Pedersen. **Non-interactive and information-theoretic secure verifiable secret sharing.** https://link.springer.com/content/pdf/10.1007/3-540-46766-1_9.pdf, 1991.

[8] A. Poelstra. **Mimblewimble.** https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.pdf, 2016.

Zero-Knowledge proof of knowledge - details

- Proof that yields nothing but its validity.
  - Alice tries to convince Bob of being in possession of some secret information.
  - Proof performed in Zero-Knowledge.

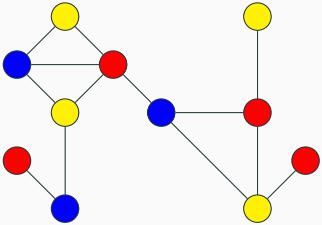

Figure 4: Graph 3-colorability problem

Zero-Knowledge proof of knowledge - details

- Proof that yields nothing but its validity.
  - Alice tries to convince Bob of being in possession of some secret information.
  - Proof performed in Zero-Knowledge.

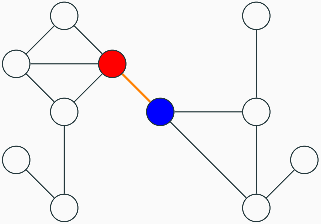

Figure 4: Graph 3-colorability problem

- Completeness.

Zero-Knowledge proof of knowledge - details

- Proof that yields nothing but its validity.
  - Alice tries to convince Bob of being in possession of some secret information.
  - Proof performed in Zero-Knowledge.

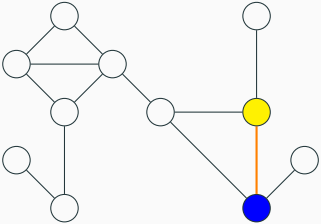

Figure 4: Graph 3-colorability problem

- Completeness.

Zero-Knowledge proof of knowledge - details

- Proof that yields nothing but its validity.
  - Alice tries to convince Bob of being in possession of some secret information.
  - Proof performed in Zero-Knowledge.

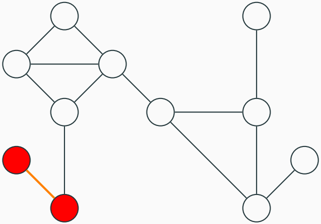

Figure 4: Graph 3-colorability problem

- Soundness.

Zero-Knowledge proof of knowledge - details

- Proof that yields nothing but its validity.
  - Alice tries to convince Bob of being in possession of some secret information.
  - Proof performed in Zero-Knowledge.

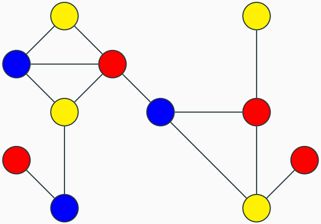

Figure 4: Graph 3-colorability problem

- Zero-Knowledgeness.

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
    2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
    3. $K_i \leftarrow s_i G - e_i Q_i$;

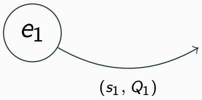

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;

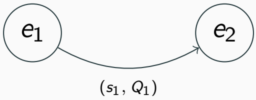

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
    2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
    3. $K_i \leftarrow s_i G - e_i Q_i$;

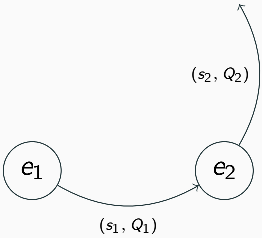

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;

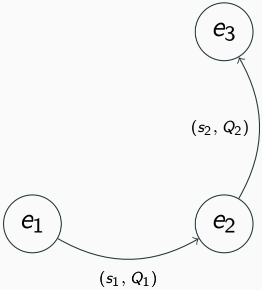

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
    2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
    3. $K_i \leftarrow s_i G - e_i Q_i$;

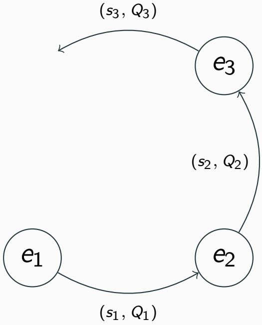

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
   1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
   2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
   3. $K_i \leftarrow s_i G - e_i Q_i$;
4. $e_{i^*} \leftarrow hash(K_{i^*-1} || m || i^*)$;

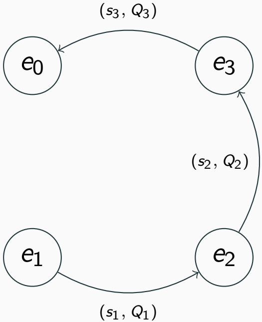

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
    1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
    2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
    3. $K_i \leftarrow s_i G - e_i Q_i$;
4. $e_{i^*} \leftarrow hash(K_{i^*-1} || m || i^*)$;
5. $s_{i^*} \leftarrow k_{i^*} + e_{i^*} q_{i^*}$;

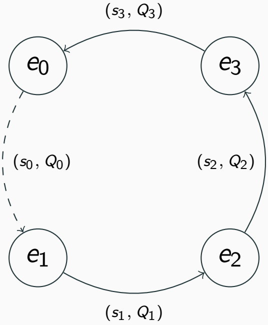

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_SIGN(m, q_{i^*}, Q_i: $0 \leq i \leq r-1$):

1. $k_{i^*} \xleftarrow{\$} \{1, \ldots, n-1\}$;
2. $K_{i^*} \leftarrow k_{i^*} G$;
3. for $i \leftarrow i^* + 1, \ldots, r-1, 0, \ldots, i^* - 1$
   1. $e_i \leftarrow hash(K_{i-1} || m || i)$;
   2. $s_i \xleftarrow{\$} \{1, \ldots, n-1\}$;
   3. $K_i \leftarrow s_i G - e_i Q_i$;
4. $e_{i^*} \leftarrow hash(K_{i^*-1} || m || i^*)$;
5. $s_{i^*} \leftarrow k_{i^*} + e_{i^*} q_{i^*}$;
6. **return** $(e_0, s_0, \ldots, s_{r-1}) =: \sigma$

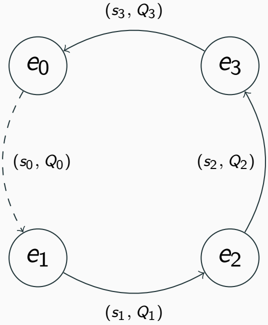

Figure 5: AOS ring signature (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
    1. $K_i \leftarrow s_i G - e_i Q_i$;

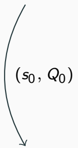

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i || m || i + 1\%r)$;

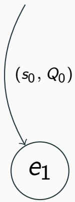

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;

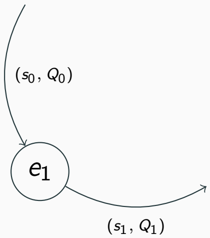

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i||m||i+1\%r)$;

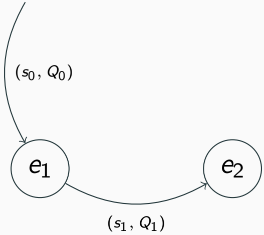

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;

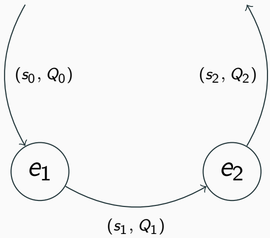

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i || m || i+1\%r)$;

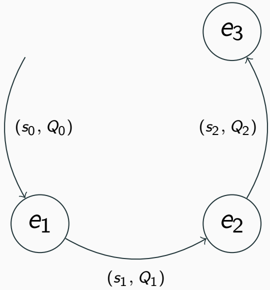

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;

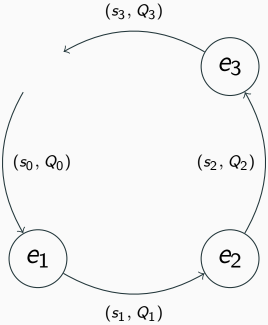

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \leq i \leq r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i || m || i + 1\%r)$;

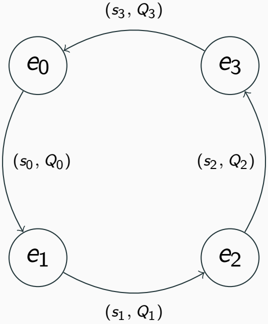

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

AOS ring signature - details of the algorithms

AOS_VERIFY(m, σ, Q_i: $0 \le i \le r-1$):

1. for $i \leftarrow 0, \ldots, r-1$
   1. $K_i \leftarrow s_i G - e_i Q_i$;
   2. $e_{i+1\%r} \leftarrow hash(K_i || m || i + 1\%r)$;
2. **if** $e_0 = 0$ or $e_0 \ge n$:
   return False;
3. **if** $e_0 = \sigma[0]$:
   return True;

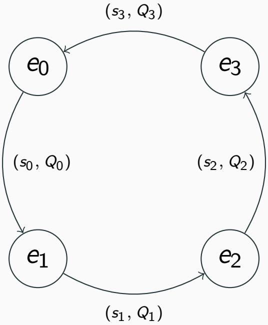

Figure 5: AOS ring signature verification (1-of-4)

Adapted from: [5]

ECDH Key Exchange protocol

- Key agreement scheme based on elliptic curve cryptography.
- Establish a shared secret between two parties over an insecure (yet authenticated) channel.

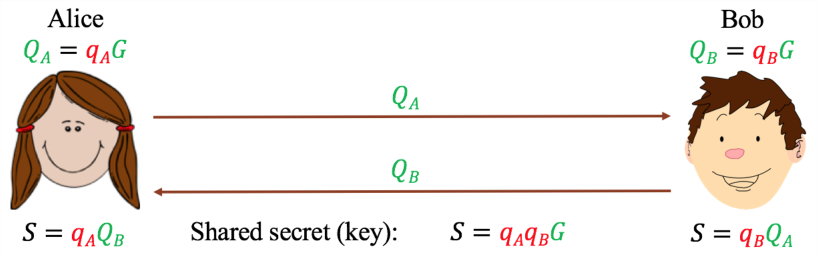

Figure 6: ECDH key exchange

- Channel authentication required to prevent *man-in-the-middle* attacks.

Sender-receiver interaction

- Encryption even prevents the receiver to know amount and blinding factor associated to each output.

Transmission of:

- amounts;
- blinding factors;
- user-selected data.

$\Rightarrow$ The transfer can happen non-interactively by running an instance of ECDH and exploiting the shared key to deduce the quantities involved.