How the hell
does Monero work?

@pwrcycle > cafecode.com/shellcon2018-monero.pdf

whois pwrcycle

Prolexic	SOC
Verisign	VIDN
Apple	SIRT
F5	Silverline
Salesforce	NetSec

---

pwrcycle on

Gmail	Reddit
Twitter	Freenode
LinkedIn	Signal

@pwrcycle > cafecode.com/shellcon2018-monero.pdf

4 Pillars of Monero's Cryptography

• Ring Signatures	Obscures the Sender
• Ring CT	Obscures the Amount
• Stealth Addresses	Obscures the Receiver
• (Kovri i2p router)	Obscures the entire

Ring Signatures = Anonymity for the Sender

Ring signatures are composed of a ring of keys and a signature from that ring. Each signature is generated with a Moner user's private key and a set of unrelated public keys.

A recipient verifying a signed transaction would not be able to tell which ring member corresponds to the sender's key that created the transaction.

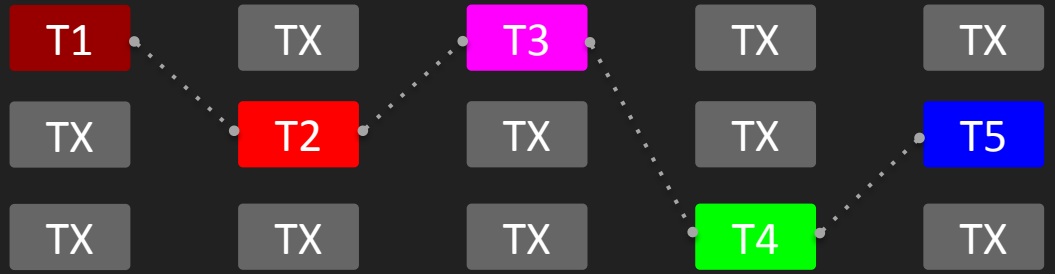
Ring Signatures = Anonymity for the Sender

Ring signatures were originally called "Group Signatures" (David Chaum and Eugene van Heyst in 1991) because they were thought of as a way to prove a signer belongs to a group, without necessarily identifying an individual.

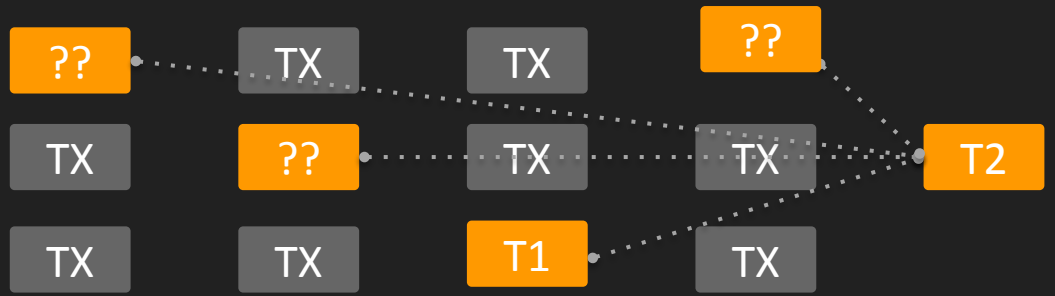
Ring Signatures allow for unforgeable, signer-ambiguous transactions that leave currency flows largelyuntraceable.

Ring Signatures = Anonymity for the Sender

Bitcoin's open ledger allows all transactions to be linked, and possibly blacklisted or confiscated.



Monero's Ring Signatures, RingCT, & Stealth Addresses make determining any previous transaction nearly impossible.



@pwrcycle > cafecode.com/shellcon2018-monero.pdf

Ring CT = Conceal of the Amount.

RingCT, implemented in Jan, 2017 (v4 of the Monero protocol).
Miners confirm blocks and transactions. Miners don’t know how much Monero is contained in each input and output, but they still need to prove the sum of input amounts equals the sum of output amounts.

The RingCT Formula

Since commitments are additive and we don't know γ, we could easily prove our inputs equal outputs to observers by making the sum of commitments to input and output amounts equal zero:[4]

$$\sum_j C_{j,in} - \sum_t C_{t,out} = 0$$

To avoid sender identifiability, Shen Noether proposed [61] verifying that commitments sum to a certain non-zero value:

$$\sum_j C_{j,in} - \sum_t C_{t,out} = zG$$

$$\sum_j (x_j G + a_j H) - \sum_t (y_t G + b_t H) = zG$$

$$\sum_j x_j - \sum_t y_t = z$$

@pwrcycle > cafecode.com/shellcon2018-monero.pdf

Stealth Addresses = Anonymity for the Receiver

The Sender automatically creates a Stealth Addresses, a random one-time addresses, for every transaction on behalf of the Receiver. The Receiver can publish one wallet address, yet all incoming payments will go to a unique addresses on the blockchain.

The Sender uses the Receiver's public key to cryptographically address the transaction so that only the Receiver can read it from the blockchain.

Stealth Addresses = Anonymity for the Receiver

1. Alice generates a random number $r \in_R \mathbb{Z}_l$, and calculates the one-time public key[2]

$$K^o = \mathcal{H}_n(rK_B^v)G + K_B^s$$

2. Alice sets K^o as the addressee of the payment, adds the value rG to the transaction data, and submits it to the network.

3. Bob receives the data and sees the values rG and K^o. He can calculate $k_B^v rG = rK_B^v$. He can then calculate $K_B'^s = K^o - \mathcal{H}_n(rK_B^v)G$. When he sees that $K_B'^s = K_B^s$, he knows the transaction is addressed to him.[3]

   The private key k_B^v is called the *view key* because anyone who has it (and Bob's public spend key K_B^s) can calculate $K_B'^s$ for every transaction output in the blockchain (record of transactions), and 'view' which ones are addressed to Bob.

4. The one-time keys for the output are:

$$K^o = \mathcal{H}_n(rK_B^v)G + K_B^s = (\mathcal{H}_n(rK_B^v) + k_B^s)G$$
$$k^o = \mathcal{H}_n(rK_B^v) + k_B^s$$

Since Monero encrypts every transaction, Bob must compute every transaction to see if it is addressed to him.

(This can make Monero wallet syncing take a while.)

@pwrcycle > cafecode.com/shellcon2018-monero.pdf

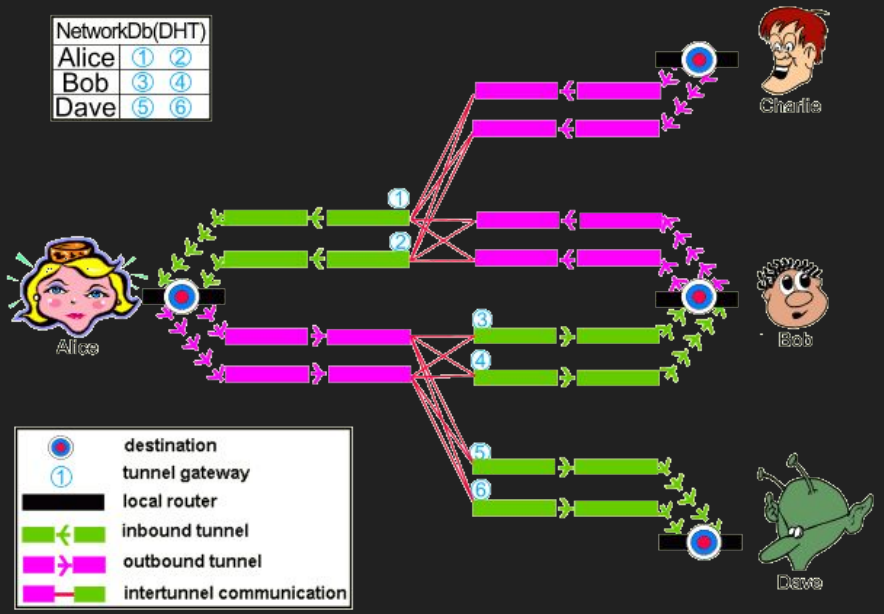
Kovri = i2p router

A lightweight and security-focused i2p router written c++. i2p is a network similar to Tor, but without entry/exit nodes or node hierarchy.

Kovri, as a router, uses i2p garlic-encryption and garlic-routing to create a private, protected overlay-network across the Internet.

Kovri = i2p router

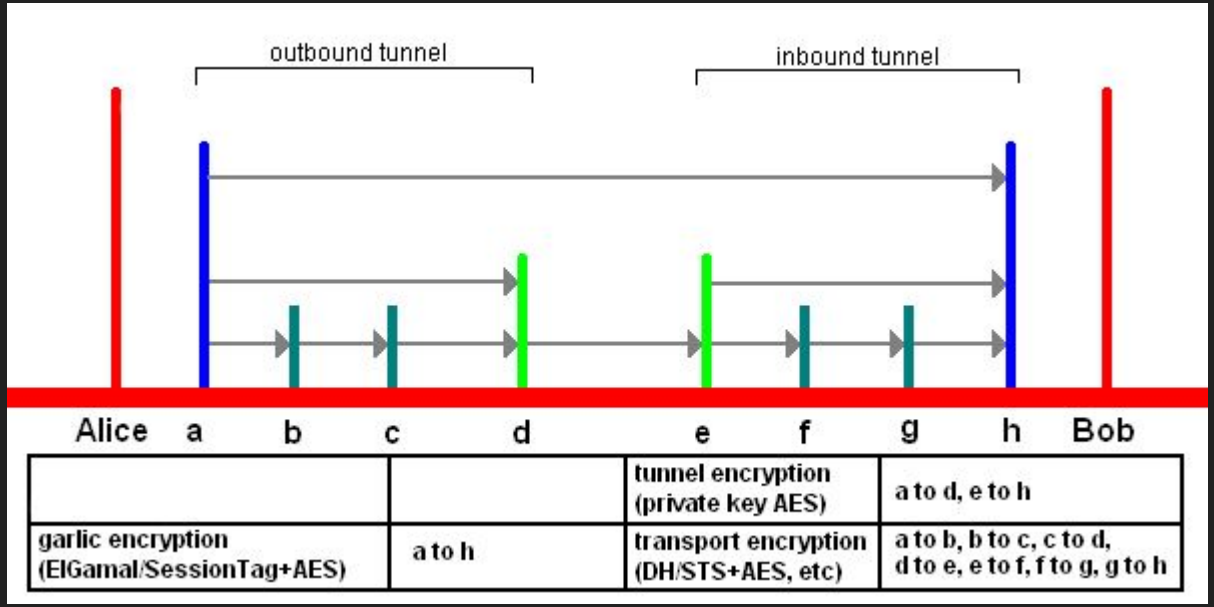
In Garlic routing, the packet is encrypted for each hop. Like Russian dolls, or a letter, inside a letter, inside a letter. The receiver does not know if the packet is destined for her, or another peer, until she decodes the packet.



@pwrcycle > cafecode.com/shellcon2018-monero.pdf

Kovri = i2p router

i2p further obfuscated user traffic by separating inbound & outbound traffic into 2 different tunnels.



		tunnel encryption (private key AES)	a to d, e to h
garlic encryption (ElGamal/SessionTag+AES)	a to h	transport encryption (DH/STS+AES, etc)	a to b, b to c, c to d, d to e, e to f, f to g, g to h

@pwrcycle > cafecode.com/shellcon2018-monero.pdf

Why are criminals adopting Monero?

Why are criminals using Monero instead of other crypto coins?

1. **Anonymity**
2. **ASIC resistance**
   Monero's focus on privacy emphasizes decentralization. This means keeping mining in the reach of average users who use commodity hardware, instead of specialized, expensive centralized ASICs.

Links

https://getMonero.org/
https://Kovri.io

https://getMonero.org/library/Zero-to-Monero-1-0-0.pdf