GOST N BLOCKCHAIN

COMPRESSED SIGNATURE AND PUBLIC KEY RECOVERY WITH GOST R 34.10-2012

by Dmitriy Bozhko and Alexey Troshichev
WHO WE ARE

ALEXEY TROSHICHEV
El Capitan
amt@ledgers.world

DMITRY BOZHKO
Crypto Core Developer
dmitry@ledgers.world

DISTRIBUTED LEDGERS
Blockchain Dev and Security
www.ledgers.world
Digital signature is to answer the following questions:

- Is the person who signed a message the same person who announces that he signed the message?
- Signed and announced messages are the same?
DIGITAL SIGNATURE: CLASSIC APPROACH

- MSG
- SIG
- PubK

DSA

- False
- True
DIGITAL SIGNATURE: NEW APPROACH

- MSG
- SIG

CECDSA
compressed

PubK
Less data to send and store: MSG + SIG + PubK > MSG + SIG

Suitable for short messages

Standard for cryptocurrency world: Bitcoin, Ethereum, Litecoin...
Legal power of digital signature

Reuse of cryptocurrency algorithms without modification
Elliptic Curve Digital Signature Algorithm (ECDSA)

GOST R 34.10-2012 (RFC7091)
Find public key \( Q \) using signature \( S = (r, s) \) as follows:

1. For \( j \) from 0 to \( h \) do the following:
   1. Let \( x = r + jn \)
   2. Convert the integer \( x \) to an octet string \( X \).
   3. Convert the octet string \( \text{02}16||X \) to an elliptic curve point \( R \). If this conversion routine outputs invalid then do another iteration of Step 1.
   4. If \( n_r \neq 0 \) then do another iteration of Step 1.
   5. Use the hash function to compute the hash value: \( H = \text{Hash}(M) \).
      If the hash function outputs invalid, output invalid and stop.
   6. Derive an integer \( e \) from \( H \).
   7. For \( k \) from 1 to 2 do the following:
      1. Compute a candidate public key as \( Q = r^{-1}(sR - eG) \)
      2. Verify that \( Q \) is the authentic public key.
         If \( Q \) is authenticated, output \( Q \) and stop.
      3. Change \( R \) to \(-R\).

2. Output invalid
<table>
<thead>
<tr>
<th></th>
<th>Sign</th>
<th>Verify</th>
<th>Recover</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ECDSA</strong></td>
<td>$s = k^{-1}(e - rd_U)$</td>
<td>$C = s^{-1}eG + s^{-1}rQ$</td>
<td>$Q = r^{-1}(sR - eG)$</td>
</tr>
<tr>
<td><strong>GOST R 34.10-2012</strong></td>
<td>$s = k^{-1}(rd_U + ke)$</td>
<td>$C = e^{-1}sG + e^{-1}rQ$</td>
<td>$Q = r^{-1}(sG - eR)$</td>
</tr>
</tbody>
</table>
PUBLIC KEY RECOVERY (GOST R 34.10-12)

Bob signs a message with a public key.

Signature $\zeta' = (r \| s \| py)$

32 bytes 32 bytes 1 byte

- $r$
- $s$
- $py$

Calculate $r$, $s$, $py$ from $\zeta'$

- $0 < r < q$
- $0 < s < q$
- $py \equiv 0 \mod 1$

Invalid $\zeta'$

$\alpha = \text{Hash}(M)$

$e = \alpha \mod q$

Calculate $R$

Calculate $r^{-1}$

$rr^{-1} = 1 \mod q$

$Q = r^{-1}(sP - eR)$

Pubkey $Q$
**COMPRESSED SIGNATURE CALCULATING (GOST R 34.10-12)**

- $d_U$ – Private key
- $M$ – Message
- $\zeta$ – Regular signature
- $p_y$ – Parity byte
- $c_{rr}$ – Error
- $\zeta'$ – Compressed signature

Diagram:

1. $d_U, M$ → $\zeta = \text{Sign}(d_U, M)$
2. $(p_y, c_{rr}) = \text{Recover}(\zeta)$
3. If $c_{rr}$
   - Yes: $\zeta' = \zeta \| p_y$
   - No: (feedback path)

Compressed signature $\zeta'$
QUESTIONS?

ALEXEY TROSHICHEV
El Capitan
amt@ledgers.world

DMITRY BOZHKO
Crypto Core Developer
dmitry@ledgers.world

DISTRIBUTED LEDGERS
Blockchain Dev and Security
www.ledgers.world
#distributedledgers
@
#zeronights