Schnorr Signatures
Digital Signatures

1976 - “New Directions in Cryptography” - Whitfield Diffie & Martin Hellman.

“Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems,... and supply the equivalent of a written signature...
A second problem, amenable to cryptographic solution, which stands in the way of replacing contemporary business communications by teleprocessing systems is authentication...
It must be easy for anyone to recognize the signature as authentic, but impossible for anyone other than the legitimate signer to produce it”

History of Digital signatures

Elgamal Signatures (1985) - \[ sk = e - kG_x d \]

ECDSA - NIST/ANSI (1997) - \[ sk = e + kG_x d \]

Schnorr Signatures (1991) - (patent expired 2008)

\[ s = k + ed \]

\[ \text{Sig}(s, kG) \]
\[ \text{Sig}(s, R) \]
Now the fun begins
Multi Signatures

\[ P_1 = d_1 \cdot G, \quad P_2 = d_2 \cdot G \]
\[ s_1 = k_1 + e \cdot d_1, \quad s_2 = k_2 + e \cdot d_2 \]
\[ s_1 + s_2 = (k_1 + k_2) + e (d_1 + d_2) \]
\[ s' = k' + e \cdot d' \]
\[ P' = (d_1 + d_2) \cdot G \]

Glossary

\[ m \] - Message.
\[ e = H(m) \]
\[ d = \text{Private Key} \]
\[ k = \text{Random nonce} \]
\[ G = \text{Generator Point} \]
Point = scalar*G = (x,y)
Public key = dG
Pay to Contract

\[ P' = P + H(P || s)G \]
\[ d' = d + H(P || s) \]
Sign to contract

\[ s = k + ed \]

\[ k' = k + H(R || c) \]

\[ R' = R + H(R || c)G \]
Anti Nonce Covert Channel

\[ R = kG \]

\[ C \]

\[ s = k' + ed \]

\[ k' = k + H(R||e) \]

**Glossary**

- \( m \) - Message.
- \( e = H(m) \)
- \( d \) - Private Key.
- \( k \) - Random nonce
- \( G \) - Generator Point.
- Point = scalar\(^*\)G = (x,y)
- Public key = dG

© Elichai Turkel
Adaptor Signatures

\[ s'_1 = k_1 + ed_1 + t \]

\[ \text{AdapSig}(s', kG, tG) \]

\[ s_2 = k_2 + ed_2 \]

\[ s' = (k_1 + k_2) + e(d_1 + d_2) \]

\[ t = s' - s_2 - s'_1 \]

\[ s_1 = s'_1 - t \]

Glossary

- \( m \) - Message.
- \( e = H(m) \)
- \( d \) - Private Key.
- \( k \) - Random nonce
- \( G \) - Generator Point.
- Point = scalar*G = (x,y)
- Public key = dG
**Atomic Swap**

\[ e = H(m) \]

- **d** = Private Key.
- **k** = Random nonce
- **G** = Generator Point.
- Point = scalar\*G = (x,y)
- Public key = dG

\[ s'_1 = k_1 + ed_1 + t \]

\[ s_2 = k_2 + ed_2 \]

\[ s'_{combined} = (k_1 + k_2) + e(d_1 + d_2) \]
Thank you and questions?