Taproot e MAST in Bitcoin

Privacy, Flessibilità e Efficienza negli Script

Valerio Vaccaro

Satoshi Spritz Connect

11 Novembre 2025

Valerio Vaccaro ∞

- 💻 Sviluppatore Bitcoin ed Esperto Hardware
- 🔥 Contributore a progetti Bitcoin open source
- ⚠️ Appassionato di hardware fai-da-te (DIY)
- Ingegnere Bitcoin e Liquid presso Blockstream

Social

- **LinkedIn** linkedin.com/in/valeriovaccaro
- **Github** github.com/valerio-vaccaro
- **Telegram** t.me/valeriovaccaro

Licenza

May this presentation inspire you to become more self-sovereign!

Meme


MERKLE TREES
LIGHTYEAR
MERKLE TREES EVERYWHERE
imgflip.com

Sommario

Contenuti della Presentazione

Introduzione a Taproot e MAST

Cos'è Taproot?

- 🚀 **Soft fork** attivato il **14 Novembre 2021** (blocco 709,632)
- 🛡️ Migliora **privacy, efficienza e flessibilità** degli script Bitcoin
- 🔒 Proposta da **Gregory Maxwell** nel 2018, sviluppata da **Pieter Wuille** e altri
- 🪙 Basato su **Schnorr signatures** e **MAST** (Merkle Abstract Syntax Tree)

Cos'è MAST?

- 🌲 **Merkle Abstract Syntax Tree**: struttura ad albero per script complessi
- 👀 Permette di **nascondere script alternativi** non utilizzati
- 🔥 Riduce drasticamente le **dimensioni delle transazioni**
- 🔒 Migliora la **privacy** nascondendo condizioni di spesa

Introduzione a Taproot e MAST

Punti Chiave

- **Privacy migliorata**: tutte le transazioni Taproot sembrano identiche
- **Efficienza**: transazioni più piccole e fee più basse
- **Flessibilità**: supporto per script complessi senza compromettere la privacy

Storia e Attivazione

Timeline dello Sviluppo

- **2018**: Gregory Maxwell propone Taproot
- **2019**: Implementazione iniziale da Pieter Wuille
- **2020**: Proposta BIP 340 (Schnorr), BIP 341 (Taproot), BIP 342 (Tapscript)
- **2021**: Lock-in al blocco 687,284 (12 Giugno)
- **14 Novembre 2021**: Attivazione al blocco **709,632**

Dettagli dell'Attivazione

Blocco di Attivazione - **Altezza**: 709,632 - **Data**: 14 Novembre 2021 - **Hash**: 00000000000000000000626d...726b

Supporto Miner - **90%+ dei miner** hanno segnalato supporto - **Soft fork** retrocompatibile - **Nessuna interruzione** della rete

Storia e Attivazione

BIP Coinvolti

- **BIP 340**: Schnorr Signatures
- **BIP 341**: Taproot (script version 1)
- **BIP 342**: Tapscript (nuovi opcodes)

Fondamenti Matematici

Curve Ellittiche e Schnorr

Schnorr Signatures: - Firma digitale più semplice ed efficiente - **Aggregazione lineare**: più firme possono essere combinate - **Non-malleability**: firme non possono essere modificate - **Batch verification**: verifica più veloce di più firme

Formula Base

Per una chiave pubblica $\mathbf{P}$ e messaggio $\mathbf{m}$:
Firma: $\mathbf{(R, s)}$ dove: - $\mathbf{R = k \cdot G}$ (punto casuale sulla curva) - $\mathbf{s = k + H(R||P||m) \cdot x}$ (scalare)
Verifica: $\mathbf{s \cdot G = R + H(R||P||m) \cdot P}$

Vantaggi rispetto a ECDSA

- ✓ **Aggregazione nativa**: più firme = una firma
- 🚀 **Verifica più veloce**: batch verification
- 🛡️ **Sicurezza dimostrabile**: riduzione a problemi matematici noti

Schnorr Signatures

Introduzione a Schnorr

- **Firma digitale** proposta da Claus-Peter Schnorr nel 1989
- **Più semplice** di ECDSA: struttura matematica più elegante
- **Sicurezza dimostrabile**: riduzione al problema del logaritmo discreto
- **Base per Taproot**: abilità di aggregare firme

Caratteristiche Principali

- ✓ **Linearità**: firme possono essere combinate matematicamente
- **Non-malleability**: firme non possono essere modificate
- **Efficienza**: verifica più veloce di ECDSA
- **Privacy**: aggregazione migliora la privacy

Aggregazione delle Firme

Proprietà fondamentale:

```
Firma 1: (R1, s1) per messaggio m1
Firma 2: (R2, s2) per messaggio m2

Firma aggregata: (R1 + R2, s1 + s2)
```

Vantaggi: - 🔥 **Multisig efficiente**: 3 firme = 1 firma aggregata - 🚀 **Riduzione dimensioni**: transazioni più piccole - 🛡️ **Privacy**: impossibile distinguere firme individuali

Schnorr Signatures

Batch Verification

Verifica simultanea di più firme:

```
Verifica sequenziale (ECDSA):
  Verifica(sig1) + Verifica(sig2) + Verifica(sig3) = 3 operazioni

Verifica batch (Schnorr):
  Verifica(sig1 + sig2 + sig3) = 1 operazione
```

Risparmio: fino a **70% più veloce** per verifiche multiple

Sicurezza di Schnorr

- **Riduzione matematica**: sicurezza basata su logaritmo discreto
- **Prova formale**: dimostrabilmente sicuro
- **Resistente a attacchi**: non vulnerabile a malleability
- **Standardizzato**: BIP 340 per Bitcoin

Come Funziona Taproot

Struttura Base

Un output Taproot può essere speso in **due modi**:

Spesa Cooperativa 🤝 - Tutti i partecipanti firmano insieme - Sembra una normale transazione P2PKH - **Massima privacy**

Spesa Non-Cooperativa ⚠️ - Rivelazione dello script alternativo - Dimostrazione del path MAST - **Privacy ridotta ma funzionale**

Output Taproot

`Output = P2TR = Pay to Taproot`
`ScriptPubKey = OP_1 <32-byte tweaked public key>`
La chiave pubblica è **tweaked** con: - $\mathbf{Q = P + H(P||script\_root) \cdot G}$ - Dove **script_root** è la root dell'albero MAST

MAST: Merkle Abstract Syntax Tree

Cos'è un MAST?

- **Albero di Merkle** che contiene tutti gli script possibili
- Solo lo **script utilizzato** viene rivelato
- Gli altri script rimangono **nascosti** nella blockchain
- Riduce le dimensioni: solo il path necessario viene incluso

Esempio Pratico

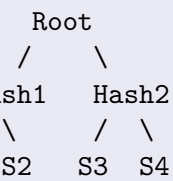
Supponiamo di avere 3 condizioni di spesa:

```
Script 1: 2-of-3 multisig (Alice, Bob, Charlie)
Script 2: Timelock 1 anno
Script 3: Hash preimage
```

Senza MAST: tutti e 3 gli script nella transazione
Con MAST: solo lo script utilizzato + proof Merkle

MAST: Merkle Abstract Syntax Tree

Costruzione dell'Albero



Per spendere con **S2**: - Includi **S2** - Includi **Hash(S1)** (sibling) - Includi **Hash2** (parent sibling) - Verifica: **Hash(Hash(S1)||Hash(S2)) = Hash1**

Esempio Pratico

Scenario: Multisig 2-of-3

Setup: - Alice, Bob, Charlie vogliono un wallet condiviso - Qualsiasi 2 di loro possono spendere - Con Taproot: sembra una normale transazione!

Spesa Cooperativa

1. Alice e Bob firmano insieme
2. Transazione sembra singlesig normale
3. Nessuno sa che è un multisig
4. Privacy massima

Esempio Pratico

Spesa Non-Cooperativa

1. Solo Alice vuole spendere
2. Rivelazione dello script multisig
3. Proof MAST per dimostrare validità
4. Transazione più grande ma funzionale

Dimensioni

Senza Taproot: - Multisig 2-of-3: ~250 bytes
Con Taproot (cooperativo): - Spesa normale: ~58 bytes (come P2PKH!)
Con Taproot (non-cooperativo): - Script + proof: ~150-200 bytes (comunque migliore)

Vantaggi della Privacy

Privacy by Default

- **Tutte le transazioni Taproot sembrano identiche**
- **Impossibile distinguere** tra:
  - Spesa semplice (single sig)
  - Multisig complesso
  - Script con timelock
  - Contratti intelligenti

Analisi della Blockchain

Prima di Taproot: - Analisti potevano identificare: - Multisig wallets - Script complessi - Pattern di spesa

Dopo Taproot: - Tutto sembra identico - Analisi blockchain molto più difficile - Privacy fungibile migliorata

Vantaggi della Privacy

Esempio Reale

```
Wallet A: Multisig 3-of-5 oppure 2-of-3 con timelock
Wallet B: Single sig (utente normale)

Senza Taproot: facilmente distinguibili
Con Taproot: identici all'analisi esterna
```

Efficienza e Dimensioni

Riduzione delle Dimensioni

Transazione Standard (P2PKH): - Input: ~148 bytes - Output: ~34 bytes - **Totale**: ~182 bytes
Transazione Taproot (cooperativa): - Input: ~58 bytes (60% più piccola!) - Output: ~43 bytes - **Totale**: ~101 bytes

Implementazione Tecnica

Script Version 1

Taproot introduce **script version 1** (Tapscript):
`OP_1 <32-byte tweaked pubkey>`
Diverso da: - **P2PKH**: `OP_DUP OP_HASH160 <20-byte hash> OP_EQUALVERIFY OP_CHECKSIG` - **P2SH**: `OP_HASH160 <20-byte hash> OP_EQUAL` - **P2WPKH**: `OP_0 <20-byte hash>`

Nuovi Opcodes

BIP 342 introduce nuovi opcodes: - ⚙ **OP_CHECKSIGADD**: per aggregazione firme - 🔑 **OP_SUCCESS**: per future estensioni - 🔒 **Miglioramenti** agli opcodes esistenti

Implementazione Tecnica

Verifica della Spesa

Path 1: Spesa Cooperativa

1. Verifica firma Schnorr aggregata
2. Se valida → spesa autorizzata
3. Nessuno script rivelato

Path 2: Spesa Non-Cooperativa

1. Rivelazione script + proof MAST
2. Verifica proof Merkle
3. Esecuzione script rivelato
4. Se valido → spesa autorizzata

Casi d'Uso

Wallet Multisig

- **Custody aziendale**: 3-of-5, 5-of-7, etc.
- **Privacy**: sembra wallet normale
- **Efficienza**: fee più basse

Lightning Network

- **Canali più efficienti**: transazioni più piccole
- **Privacy migliorata**: canali indistinguibili
- **Eltoo**: protocollo migliorato con Taproot

Casi d'Uso

Smart Contracts

- **Contratti complessi**: senza rivelare logica
- **Timelocks**: nascosti fino all'uso
- **Condizioni multiple**: tutte nascoste

Escrow e Custody

- **Servizi di custodia**: privacy per clienti
- **Escrow**: condizioni nascoste
- **Trust minimizzato**: senza compromettere privacy

⚠ Limitazioni e Considerazioni

Limitazioni Attuali

- 👀 **Adozione**: non tutti i wallet supportano ancora
- ⚙ **Compatibilità**: alcuni servizi non ancora aggiornati
- ⚠ **Spesa non-cooperativa**: rivelazione parziale di privacy

Considerazioni di Sicurezza

- 🛡 **Schnorr**: matematicamente sicuro ma nuovo (nuove implementazioni)
- 🔥 **Implementazione**: richiede attenzione ai dettagli
- 🔒 **Audit**: codice open source e verificato

⚠ Limitazioni e Considerazioni

Best Practices

- ✓ **Usa spesa cooperativa** quando possibile
- 🔑 **Testa script** prima di usare in produzione
- 🛡 **Verifica wallet** supportano Taproot

Adozione e Statistiche

Statistiche di Adozione

Novembre 2021 - Attivazione: - 🚀 **0%** delle transazioni
Gennaio 2025: - 🔥 **~15-20%** delle transazioni - 🪙 Crescita costante mese su mese - 🛡️ Supporto da wallet principali

Wallet che Supportano

- ✓ **Bitcoin Core**: supporto nativo
- ✓ **Electrum**: supporto completo
- ✓ **Sparrow Wallet**: supporto completo
- ✓ **BlueWallet**: supporto completo
- ⚠️ Alcuni wallet ancora in sviluppo

Adozione e Statistiche

Miner e Nodi

- 🔥 **100% dei miner** supportano Taproot
- ⚙️ **Maggior parte dei nodi** aggiornati
- 🛡️ **Retrocompatibilità**: nodi vecchi ancora funzionano

Implementazione Pratica

```
# Genera chiave privata
private_key = bytes(os.urandom(32))
public_key = wally.ec_public_key_from_private_key(private_key)

# Crea script alternativo (multisig)
script = create_multisig_script([pub1, pub2, pub3])

# Costruisci albero MAST
script_root = build_mast_tree([script])

# Tweaka la chiave pubblica
tweaked_pubkey = taproot_tweak_pubkey(public_key, script_root)

# Crea output Taproot
output = create_p2tr_output(tweaked_pubkey)
```

Implementazione Pratica

Esempio: Spendere Output Taproot

Spesa Cooperativa:

```
# Tutti i partecipanti firmano
signatures = [sign1, sign2, sign3]
aggregated_sig = aggregate_signatures(signatures)

# Crea transazione
tx = create_transaction(input, output, aggregated_sig)
```

Spesa Non-Cooperativa:

```
# Rivelazione script + proof
script = get_script_from_mast(script_index)
merkle_proof = get_merkle_proof(script_index)

# Crea witness
witness = [script, merkle_proof, ...]
```

Futuro e Sviluppi

Potenziali Miglioramenti

- **Graftroot**: estensione per script più complessi
- **Cross-input aggregation**: aggregazione tra input
- **Scriptless scripts**: logica senza script espliciti

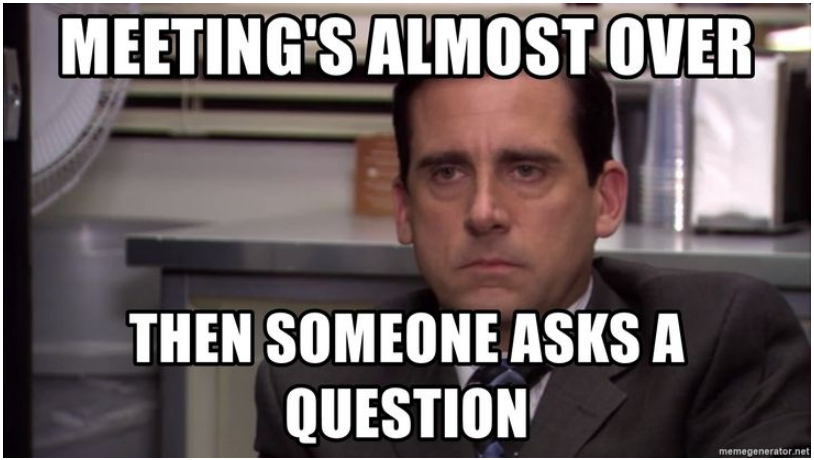
Integrazione con Altri Protocolli

- **Lightning Network**: canali più efficienti
- **DLCs**: Discrete Log Contracts migliorati
- **Sidechains**: integrazione con Liquid, Rootstock

Ricerca Attiva

- **Zero-knowledge proofs**: integrazione futura
- **Privacy avanzata**: tecniche aggiuntive
- **Scalabilità**: miglioramenti continui

? Domande


MEETING'S ALMOST OVER
THEN SOMEONE ASKS A QUESTION
memegenerator.net

Bibliografia

- Gregory Maxwell, et al. “Taproot: Privacy Preserving Switchable Scripts” (2018)
- Pieter Wuille, Jonas Nick, Tim Ruffing. “BIP 340: Schnorr Signatures” (2020)
- Pieter Wuille, et al. “BIP 341: Taproot” (2020)
- Pieter Wuille, et al. “BIP 342: Validation of Taproot Scripts” (2020)
- Andrew Poelstra. “Mimblewimble and Scriptless Scripts” (2016)
- Russell O’Connor, et al. “Graftroot: A Generalization of Taproot” (2018)

Progetto Satoshi Spritz

- 💻 Federazione di gruppi locali di Bitcoiner
- 🎓 Eventi gratuiti e privacy oriented
- 🤖 BITCOIN ONLY
- 🔧 Satoshi Spritz Connect online settimanale
- 📖 Orientato all'apprendimento della self-sovereign
- 🍕 Tutte le settimane un evento online -> Satoshi Spritz Connect

Links

- satoshispritz.it
- t.me/SatoshiSpritzConnect

Officine Bitcoin

- Comunità Italiana di Bitcoiners, totalmente gratuita
- BITCOIN ONLY
- Focus su educazione e sviluppo di progetti
- Progetti:
  - Sviluppo nodi Bitcoin
  - Uso di Hardware Wallet
  - Filosofia open source
  - Installazione di Debian
  - Mnemoniche & Dadi
  - . . . e molto altro

Links

- officinebitcoin.it