Midterm Exam

The midterm exam will be in class, **Wednesday, 21 October**. The midterm will be closed-resources, but is meant to test understanding, not memorization.

The midterm covers: Classes 1-15, Checkups 1-2, Problem Sets 1-2, and the assigned readings: Chapters 1, 2, 3, and 5 from *Bitcoin and Cryptocurrency Technologies*; Chapters 1, 2, 3, 4, 6, and 7 from *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*; Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*; Ittay Eyal and Emin Gün Sirer, *Majority is not Enough: Bitcoin Mining is Vulnerable*.

The first two questions on the midterm will ask you to comment on the technical validity of some statements in the Congressional Research Service report (in the first 8 pages, not the policy issues).

Extra Office Hours. There will be some additional office hours this week:

- Monday, 5-6:30pm (Ori, Rice 442)
- Tuesday, 2-3:30pm (Dave, Rice 507) **(Added)**
- Tuesday, 3:30-4:30pm (Samee, Rice 442) **(Added)**
- Wednesday, 3:30-4:30pm (Samee, Rice 442)
- Thursday, 2:30-3:30 (Dave, Rice 507)

Scripting Transactions

Download the slides

Recall from class 12: Transaction outputs in bitcoin are protected by *locking scripts*, and must be unlocked by *unlocking scripts*. A transaction output is not unlocked unless an unlocking script is provided such that the result of executing the unlocking script, followed by executing the locking script, is a stack with value True on top.

`OP_IF` *statements* `OP_ENDIF` - If the top of the stack is **1**, executes *statements*. Otherwise does nothing.

`OP_CHECKSIG` - Pops two items from the stack, *publickey* and *sig*. Verifies the entire transaction (known from node state, not the stack) using the *publickey* and *sig*. If the signature is valid, push **1**; otherwise, **0**.

```
OP_1 OP_DUP OP_ADD OP_DUP OP_SUB OP_VERIFY
```

The most common locking script (send to public address):

```
OP_DUP
OP_HASH160
OP_DATA20 (bitcoin address)
OP_EQUALVERIFY
OP_CHECKSIG
```

What must be on the stack for the locking script to succeed (end with **1** on top of stack)?

`OP_HASH160`
20-byte hash
`OP_EQUAL`

What must be on the stack for the locking script above ("Pay-to-Script-Hash") to succeed?

According to Most Popular Transaction Scripts (analysis of all transactions in first 290,000 blocks), the ninth most popular script is:

`OP_RETURN`
`OP_DATA_40`

What must be on the stack for the `OP_RETURN OP_DATA_40` locking script to succeed (end with **1** on top of stack)? (Trick question: what happens to the coin protected by this locking script?)

BTCD Code

Type: Script is the virtual machine the executes scripts (note that it has two Stacks)

Execute a script: Execute
Execute one instruction: Step

Opcodes: exec function executes one instruction

Some interesting opcode implementations: OP_IF, OP_RETURN

Bitcoin Core Code

script/interpreter.cpp, OP_DUP, Crypto, OP_CHECKSIG

Links

Script Playground

Some interesting things you can do with bitcoin scripts:
Contracts (see also Nick Szabo's *Formalizing and Securing Relationships on Public Networks*
Secure Multiparty Computations (to implement lotteries)

The `OP_RETURN`/pasted script execution bug doesn't even make this list of *The 9 Biggest Screwups in Bitcoin History*.