8-21-2024

Crypto Currency Exchange and Mining Excel Simulations

Tom Arnold
University of Richmond, tarnold@richmond.edu

Joseph Farizo
University of Richmond, jfarizo@richmond.edu

Jonathan M. Godbey

Recommended Citation

Arnold, Tom, C., Joseph Farizo, Jonathan M. Goodbye. "Crypto Currency Exchange and Mining Excel Simulations," University of Richmond Robins School of Business, (2024): 1-16.

Crypto Currency Exchange and Mining Excel Simulations

Tom Arnold, CFA, CIPM
The Robins School of Business
University of Richmond
Department of Finance
102 UR Drive
Richmond, VA 23173
tarnold@richmond.edu
O: 804-287-6399
F: 804-289-8878

Joseph Farizo
The Robins School of Business
Department of Finance
102 UR Drive
University of Richmond, VA 23173
O: 804-289-8565
F: 804-289-8878
jfarizo@richmond.edu

Jonathan M. Godbey
Robinson College of Business
Georgia State University
Department of Finance
35 Broad St
Atlanta, GA 30303
jgodbey@gsu.edu
O: 404-413-7328
F: 404-413-7312

Please do not cite without permission

August 21, 2024

Crypto Currency Exchange and Mining Excel Simulations

The mathematics underlying blockchain-based cryptocurrencies is beyond the scope of most undergraduate finance programs. However, students should understand the intuition behind blockchain so that they might better understand how to apply this technology to future cases. In this paper, we develop a mathematically simple digital signature example and a mathematically simple proof-of-work simulation for classroom use.

Keys:
Digital signature: mathematical relationship between private and public key, one-way function

Proof-of-work: random guessing, faster guessing has a cost, benefit of winning

INTRODUCTION

Cryptocurrencies have become a very popular topic in the finance classroom for many reasons. However, the complicated mathematics underlying blockchain technology is beyond the scope of most undergraduate courses in finance. Students need to have some intuition regarding this technology to understand and create new usees for blockchain. In this paper, we develop a mathematically simple digital signature example and a mathematically simple proof-of-work simulation for classroom use.

The mechanics of how a crypto exchange works are not easily illustrated nor demonstrated. The lack of a dealer or an exchange (i.e. a third party) to process or "clear" trades is very different from trading traditional securities. Crypto trading utilizes blockchain technology that records trades based on the consensus of the participants rather than a centralized exchange. Any entity can record a trade on the blockchain when receiving consensus[1] (called "proof-of-work" in the case of Bitcoin). These entities that compete to record the trades are called "miners."

In the case of Bitcoin, a miner receives newly issued Bitcoin based on a schedule as compensation for recording the trade (i.e., for updating the blockchain). The Bitcoin compensation schedule decreases the amount of new Bitcoin issued through time slowly by halving the compensation every 210,000 blocks, which occurs roughly every four years. Based on this schedule, no new Bitcoin will be created after the year 2140.

In this presentation, we provide a simulation in Excel to demonstrate "proof-of-work" for a transaction that requires encryption. This simulation uses relationships between Fibonacci numbers to create the encryption. A second simulation demonstrates

[1] While proof-of-work is the most well-known consensus protocol, many others, like proof-of-stake, exist.

the competition between miners for clearing a crypto currency trade based on the mechanism used by Bitcoin. The simulation demonstrates attempts – known as “mining” or “hashing” – by four different miners to successfully record a trade over a sixty second interval. Each miner must decide an allocation of fixed costs that increases the “frequency of attempts” to record the trade (e.g., generate an attempt each second or every two seconds, etc.) by lowering the variable cost per attempt. A higher fixed cost investment will make a miner more competitive, but that does not guarantee success. Further, a lower fixed cost investment may still generate enough compensation to make mining profitable.

Section 1 presents the “proof-of-work” with encryption simulation. Section 2 presents the crypto miner simulation. Section 3 concludes.

SECTION 1: Digital Signature Simulation

Participants in cryptocurrency transactions, which use permissionless blockchains, are anonymous. All transactions are recorded on a distributed ledger. The obvious problem arises – how can everyone in the community trust the authenticity of an anonymous message? The solution is a digital signature.

Digital signatures are one-way functions. The message sender has a public key, also called a verification key, and a private key, also called a signature key. Students are often told that there is a mathematical relationship between the keys but the exact nature of that relationship is not explained.

Consider the following simple and easily breakable digital signature scheme.

Alice chooses her private signing key to be $S_k(f_1,f_3,f_4) = S_k(3,8,13)$ and her public verification key, P_k, to be $f_4 = 13$. The community knows that the authentication equation is:

$$S_{m2}(P_k - S_{m2}) - \frac{S_{m1}}{m} = |1|$$

They do not know the values of Alice’s signature S_{m1}, S_{m2}. For a message to be authenticated it must include the message, m, and the values for S_{m1} and S_{m2} that make the authentication equation hold. Suppose that the message is m = 4. We have:

$$S_{m2}(13 - S_{m2}) - \frac{S_{m1}}{4} = |1|$$

If no one in the community has access to a search algorithm like Solver in Excel, the patience for trial and error, or the ability to recognize the somewhat hidden mathematical relationship between the signing and public keys, this scheme will work. Remember that only Alice knows her private signing key. No one else knows anything about it, except that it has a mathematical relationship to the signed messages and the public verification key.

The mathematical relationship is as follows. Alice picks four consecutive Fibonacci numbers, f_1, f_2, f_3, and f_4. Let them be 3, 5, 8, and 13. Her private signing key is $S_k(f_1,f_3,f_4)=S_k(3,8,13)$. She broadcasts her public key, $P_k = 13$, which is f_4 and the message m=4.

$S_{m1} = mf_1f_4$ and $S_{m2} = f_3$. So, $S_{m1} = 4\times3\times13=156$ and $S_{m2} = 8$. Since it is a property of Fibonacci numbers that $f_1f_4 - f_2f_3 = |1|$, we know that the authentication formula is:

$$S_{m2}(P_k - S_{m2}) - \frac{S_{m1}}{m} = |1|$$

Alice can produce a valid signed message but no one else can in this community of limited computing power and mathematical skill. One can repeat this example with any different value for m and any four consecutive Fibonacci numbers.

In the real world an observant mathematician would soon realize that the public verification key is always a Fibonacci number and that having an authentication equation equal to the absolute value of 1 looks suspiciously like $f_1f_4 - f_2f_3$. Even without a search algorithm to back into S_{m1} and S_{m2} this scheme would fail.

Validating a transaction requires a "public key" for both the sender and the recipient of the crypto currency, a "private key" known only to the individual trader, and an identifier and amount for the crypto currency. To simplify the transaction, we identify the crypto currency by a number, "12564" or referred to as "COIN#," and assume the trade is for one unit of the crypto currency.

To provide encryption, we employ Fibonacci numbers, where f(N) is the Nth Fibonacci number:

f(1) = 2

f(2) = 3

f(3) = 5 = f(2) + f(1)…f(N) = f(N – 1) + f(N – 2)

The first one hundred Fibonacci numbers are generated in the spreadsheet (see Figure 1) in columns H and I on the "DIGITAL SIGNATURE" worksheet.

To generate a random public key for the crypto currency sender, the spreadsheet generates a random whole number between 4 and 50, call this value "X," to produce a public key of f(X). Similarly, a random public key for the crypto currency recipient emerges from a random whole number between 54 and 100, call this value “Y,” to produce a pubic key of f(Y).

Each participant has two private keys: f(X – 1) and f(X – 3) for the sender and f(Y – 1) and f(Y – 3) for the recipient. To verify the sender in the transaction, the following relationship must be correct (recall COIN# = 12564) in Figure 1:

$$\{(\text{COIN\#}) \times [f(X) - f(X-3)]\} \div \{(\text{COIN\#}) \times [f(X-1) - f(X-3)]\} - 2 = 0 \qquad (1)$$

In the numerator: expand f(X) and then expand f(X – 1)

$$[f(X) - f(X-3)] = f(X-1) + f(X-2) - f(X-3)$$

$$= f(X-2) + f(X-3) + f(X-2) - f(X-3) = 2 \times f(X-2) \qquad (2)$$

In the denominator: expand f(X – 1)

$$[f(X-1) - f(X-3)] = f(X-2) + f(X-3) - f(X-3) = f(X-2) \qquad (3)$$

Inserting equations (2) and (3) into equation (1) demonstrates the result.

Similarly, to verify the recipient in the transaction, the following relationship must be correct:

$$\{(\text{COIN\#}) \times [f(Y) - f(Y-3)]\} \div \{(\text{COIN\#}) \times [f(Y-1) - f(Y-3)]\} - 2 = 0 \qquad (4)$$

The final part of the encryption is to authenticate the transaction by identifying the COIN# for the sender and recipient from each participant’s public key and one of their private keys. If COIN# matches, the transaction is legitimate between two approved participants. For the sender:

$$\{(\text{COIN\#}) \times [f(X) - f(X-3)]\} \div \{\text{Public key} - \text{Second Private key}\} = \text{COIN\#} \qquad (5)$$

Recall, the public key is f(X) and the second private key is f(X – 3).

Similarly, for the recipient:

$$\{(\text{COIN\#}) \times [f(Y) - f(Y-3)]\} \div \{\text{Public key} - \text{Second Private key}\} = \text{COIN\#} \qquad (6)$$

Again, recall, the public key is f(Y) and the second private key is f(Y – 3). Assuming all three levels of legitimizing the transaction are successful (see Figure 1), a miner can start attempting (or hashing) the transaction to record into the blockchain.

Figure 1: Digital Signature Encryption Simulation

	A	B	C	D	E	F
1	Coin Number:	12564				
2						
3	Sender:					
4	Public Key:	701408733	f(X)	NUMERATOR:	6.173215E+12	(COIN#)*[f(X) – f(X – 3)]
5	Private Siganture-1:	433494437	f(X – 1)	DENOMINATOR:	3.36608E+12	(COIN#)*[f(X - 1) – f(X – 3)]
6	Private Signature-2:	165580141	f(X – 3)	Proof:	SUCCESS	NUM/DOM – 2 = 0
7						
8	Recipient:					
9	Public Key:	1.77998E+18	f(Y)	NUMERATOR:	1.70843E+22	(COIN#)*[f(Y) – f(Y – 3)]
10	Private Siganture-1:	1.10009E+18	f(Y – 1)	DENOMINATOR:	8.54216E+21	(COIN#)*[f(Y - 1) – f(Y – 3)]
11	Private Signature-2:	4.20196E+17	f(Y – 3)	Proof:	SUCCESS	NUM/DOM – 2 = 0
12						
13						
14	Public View:					
15						
16	Sender Public Key:	701408733				
17	Recipient Public Key:	1.77998E+18				
18	Sender Coin:	12564				
19	Recipient Coin:	12564				
20	Transaction:	SUCCESS				

Figure 1: Digital Signature Encryption Simulation (continued)

	D	E	F	G	H	I
1					Index:	Fibonacci:
2					1	2
3					2	3
4	NUMERATOR:	6.173215E+12	COIN*[f(X) – f(X – 3)]		3	5
5	DENOMINATOR:	3.36608E+12	COIN*[f(X - 1) – f(X – 3)]		4	8
6	Proof:	SUCCESS	NUM/DOM – 2 = 0		5	13
7					6	21
8					7	34
9	NUMERATOR:	1.70843E+22	COIN*[f(Y) – f(Y – 3)]		8	55
10	DENOMINATOR:	8.54216E+21	COIN*[f(Y - 1) – f(Y – 3)]		9	89
11	Proof:	SUCCESS	NUM/DOM – 2 = 0		10	144
12					11	233
13					12	377

Cell formulas are available in the Appendix, hit the "F9" key to refresh the simulation

A copy of this spreadsheet is available at: https://scholarship.richmond.edu/finance-faculty-publications/XX/

For classroom purposes, the instructor may want to use a white font to hide portions of the simulation, perhaps only showing the students the COIN#, the public keys, and that the three levels of legitimization are successful. The students can attempt to steal the crypto currency by trying to unravel the private keys and legitimization tests. As guesses, informed or not, are entered into the private key cells, the simulation is programmed to determine the success or failure of these attempts.

SECTION 2: Crypto Currency Miner Simulation

Distributed ledger technology relies on a community of participants to verify messages and agree to add them to the ledger. The most popular consensus protocol is proof-of-work in which participants solve a puzzle and the winner receives a reward. In the case of bitcoin, the reward is newly minted bitcoin and transaction fees paid by the message sender. The puzzle is a game of random chance, like flipping coins with the winner being the first to flip N heads in a row. If it were a game of skill, only the skilled players would win. A game of chance encourages everyone to play.

A proof-of-work protocol (in the case of Bitcoin) this is called "mining". We develop a simulation that shows the tradeoffs of the cost of the effort of mining versus the reward for successfully solving the puzzles.

This simulation follows the work of Pritzker (2019). The transaction information provided by a miner to potentially update the blockchain is in the form of 256-bit block (a "bit" is a binary number that is either 0 or 1). Each block of information is applied to a "hashing" function that generates a 64-digit number based on the 256 bits within the block. If the 64-digit number is below a target number, the miner is permitted to update the blockchain for a certain amount of newly created crypto currency as compensation based on a schedule that lasts through the year 2140.

The hashing function provides a unique 64-digit number for an identical block, however, reversing the 64-digit number to reproduce the block is so complex that it is not worth trying to do. Consequently, if the miner is unsuccessful initially, numbers can be added (called “nonces”) to the end of transaction information to generate a new 64-digit number. The miner keeps adding nonces to make new attempts for updating the blockchain as fast as possible to “win” the right to record the transaction. The repeated attempts to record the transaction onto the blockchain is often called “mining” or “hashing.”[2]

To capture how mining or hashing works, we create an Excel file in which a given miner A, B, C, or D, chooses a fixed cost amount that increases the ability to make more frequent attempts to win the right to update the blockchain by lowering the variable cost per attempt. The fixed cost is allocated for mining over a 60-second period based on the following schedule presented in Figure 2.

Figure 2: Mining Fixed and Variable Cost Schedule

Panel A: Fixed to Variable Cost Schedule	
Fixed Cost (FC) per 60 Seconds:	**Associated Variable Cost per Attempt:**
FC ≥ $5.00	$0.04
$5.00 > FC ≥ $4.00	$0.07
$4.00 > FC ≥ $3.00	$0.10
$3.00 > FC ≥ $0.00	$0.20
Panel B: Variable Cost per Attempt to Frequency of Attempts	
Variable Cost per Attempt:	**Seconds per Attempt:**

[2] Technically, the ‘nonce’ is modified within the block header, generating a new hash value. This is repeated with different nonce values until a hash meeting the target criteria is found.

VC > $0.11	3 seconds per attempt
$0.11 ≥ VC > $0.07	2 seconds per attempt
$0.07 > VC	1 second per attempt

In viewing the schedule, one can see the trade-offs. For example, a fixed cost of $5.00 lowers the variable cost to $0.04 per attempt that can occur every second versus a fixed cost investment of $4.00 with a variable cost per attempt of $0.07 that can also occur every second. If it takes 34 seconds to reach a successful attempt, a $5.00 fixed cost has a total cost of $6.36 ($5.00 + 34 × $0.04) and a $4.00 fixed cost has a total cost of $6.38 ($4.00 + 34 × $0.07). A $5.00 fixed cost is better if more than 33 attempts are required to be successful.

Moving on from costs, a successful attempt for recording the transaction in the blockchain in the simulation generates compensation of $100.00. Consider the $100.00 to be the current value of the crypto currency.

Next, instead of having blocks of transaction information applied to a hashing function that generates an effectively random 64-digit number, each miner's attempt is assigned a random whole number between 0 and 10,000. If the value assigned to the miner is equal to or below the target value (set at 50, i.e. a 0.50% probability of being successful), the attempt is considered successful. Figure 3 illustrates this simulation and tabulates the cost of the four miners. The simulation is for 60 seconds, and success by one of the four miners is not guaranteed.

Figure 3: Crypto Mining Simulation

	A	B	C	D	E	F
1	**MAXIMUM:**	10000				
2	**MINIMUM:**	0				
3	**BARRIER:**	50.00				
4	**PROBABILITY OF SUCCESS:**	0.5000%				
5	**CRYPTO-VALUE:**	$100.00				

6						
7						
8	**MINER:**	A	B	C	D	
9	**FIXED COST:**	$5.00	$3.00	$1.50	$4.00	
10	**VARIABLE COST:**	$0.04	$0.10	$0.20	$0.07	
11	**SECONDS per ATTEMPT:**	1	2	3	1	
12	**COST:**	$5.32	$3.40	$1.90	$4.56	
13	**PROFIT:**	$(5.32)	$96.60	$(1.90)	$(4.56)	
14						
15	**TIME (SECONDS):**					**SUCCESS:**
16	1	F/A			F/A	
17	2	F/A	F/A		F/A	
18	3	F/A		F/A	F/A	
19	4	F/A	F/A		F/A	
20	5	F/A			F/A	
21	6	F/A	F/A	F/A	F/A	
22	7	F/A			F/A	
23	8	F/A	SUCCESS		F/A	1
24	9					
25	10					

Cell formulas are available in the Appendix, hit the "F9" key to refresh the simulation

A copy of this spreadsheet is available at: https://scholarship.richmond.edu/finance-faculty-publications/XX/

Pressing the "F9" key refreshes the simulation. Notice that there are many times when there are no successful attempts over sixty seconds, and that Miners A and D tend to be the most successful of the four miners over successive simulation trials because they can generate an attempt every second.

However, Miners B and C are successful periodically, and both can have lower total expenses relative to Miners A and D due to having lower fixed costs, depending on how long it takes for a successful attempt. Consequently, a good classroom exercise is to run the simulation twenty times (i.e. 20 minutes of "transaction time" or 20 trials) and determine which miner is the most profitable and whether all four miners are profitable. Next, raise the value of the crypto currency to $200.00 and run the exercise again. This type of exercise illustrates the reason for entities to try to be crypto miners or crypto hashers because of the possible rewards for even very infrequent successful attempts.

Another exercise is raising and lowering the target value. Raising it increases the probability of success, but also favors Miners A and D. How high or low the target value should be is an interesting classroom discussion. In a sense, it depends on trying to find a level that does not over-encourage nor discourage mining. Bid-ask spreads have a similar effect.

SECTION 3: Conclusion

The mechanisms employed by crypto currency markets are very different from traditional security markets. Instead of dealers and a centralized exchange, crypto markets use a blockchain to record transactions, and the blockchain is updated by miners or hashers, who could technically be anybody with the requisite computer resources.

We provide two simulations to demonstrate how encryption can work to legitimate a transaction and how miners compete to update the blockchain with a given transaction. Classroom exercises can include trying to steal crypto currency by "breaking" the three levels of legitimization for a transaction, mining for crypto currency as a reward for updating the blockchain or changing the reward parameters to encourage more miners to compete. Students can contrast how these items work relative to exchanges and discuss which marketplace may be more secure and stable.

APPENDIX:

Figure 1 cell formulas:

Assigning Fibonacci numbers to the sender’s public and two private keys

CELL B4: =VLOOKUP(RANDBETWEEN(4, 50), $H$2:$I$101, 2, FALSE)

CELL B5: =VLOOKUP(XLOOKUP($B$4,$I$2:$I101, $H$2:$H101, “N/A”, 0, 1) – 1, $H$2:$I$101, 2, FALSE)

CELL B6: =VLOOKUP(XLOOKUP($B$4,$I$2:$I101, $H$2:$H101, “N/A”, 0, 1) – 3, $H$2:$I$101, 2, FALSE)

Assigning Fibonacci numbers to the recipient’s public and two private keys

CELL B9: =VLOOKUP(RANDBETWEEN(54, 100), $H$2:$I$101, 2, FALSE)

CELL B10: =VLOOKUP(XLOOKUP($B$9,$I$2:$I101, $H$2:$H101, “N/A”, 0, 1) – 1, $H$2:$I$101, 2, FALSE)

CELL B11: =VLOOKUP(XLOOKUP($B$9,$I$2:$I101, $H$2:$H101, “N/A”, 0, 1) – 3, $H$2:$I$101, 2, FALSE)

First level of legitimization (sender):

CELL E4: = B1 * (B4 – B6)

CELL E5: = B1 * (B5 – B6)

CELL E6: =IF(ROUND(E4/E5 – 1,2) = 0, “SUCCESS”, “FAILED”)

Second level of legitimization (recipient):

CELL E9: = B1 * (B9 – B11)

CELL E10: = B1 * (B10 – B11)

CELL E11: =IF(ROUND(E9/E10 – 1,2) = 0, “SUCCESS”, “FAILED”)

Third level of legitimization (COIN#):

CELL B16: = B4

CELL B17: = B9

CELL B18: = E4 / (B4 – B6)

CELL B19: = E9 / (B9 – B11)

CELL B20: =IF(AND(E6 = “SUCCESS”, E11 = “SUCCESS”, B18 – B19), “SUCCESS”, FAILED”)

Generating Fibonacci numbers:

CELL H2: 1. CELL H3: = 1 + H2...copy downward through CELL H101

CELL I2: 2, CELL I3: 3, CELL I4: = I2 + I3…copy downward through CELL I101

Figure 3 cell formulas:

Note: The expression “ “ are double quotes around a blank line space

CELL B4: = B3 / B1

Determine miner’s variable cost:

CELL B10: =IF(B9 > = 5, 0.04, IF(B9 > = 4, 0.07, IF(B9 > = 3, 0.10, 0.20)))
Copy to cells C10, D10, and E10

Determine miner’s seconds per attempt:

CELL B11: =IF(B10 < = 0.07, 1, IF(B10 < = 0.11, 2, 3))
Copy to cells C11, D11, and E11

Determine miner’s (total) cost:

CELL B12: = B9 + B10*(MAX($A$16:$A$75) – COUNTIF(B16:B75, “ “))
Copy to cells C12, D12, and E12

Determine miner’s profit:

CELL B13: =IF(COUNTIF(B16:B75, “SUCCESS”) > 0, $B$5 / SUM($F$16:$F$75) – B12, -B12)
Copy to cells C13, D13, and E13

Time index:

CELL A16: 1, CELL A17: = 1 + A16… copy downward through CELL A75

Assignment of random value for mining/hashing simulation:

CELL B16: =IF(MOD($A16, B$11) = 0, IF(RANDBETWEEN($B$2, $B$1) - $B$3 > 0, “F/A”, “SUCCESS”), “ “)
Copy to cells C16, D16, and E16

CELL B17: =IF(AND(MOD($A17, B$11) = 0, $F$16 = “ “), IF(RANDBETWEEN($B$2, $B$1) - $B$3 > 0, “F/A”, “SUCCESS”), “ “)
Copy to cells C17, D17, and E17

CELL B18: =IF(AND(MOD($A18, B$11) = 0, SUM($F$16:$F17) = 0), IF(RANDBETWEEN($B$2, $B$1) - $B$3 > 0, “F/A”, “SUCCESS”), “ “)
Copy to cells C18, D18, and E18

Copy cells B18, C18, D18, and E18 downward through cells B75, C75, D75, and E75

Recording mining/hashing success:

CELL F16: =IF(COUNTIF(B16:E16, “SUCCESS”) > 0, COUNTIF(B16:E16, “SUCCESS”), “ “)

CELL F17: =IF(F16 = “ “, IF(COUNTIF(B17:E17, “SUCCESS”) > 0, COUNTIF(B17:E17, “SUCCESS”), “ “), “ “)

CELL F18: =IF(SUM($F$16:F17) = 0, IF(COUNTIF(B18:E18, “SUCCESS”) > 0, COUNTIF(B18:E18, “SUCCESS”), “ “), “ “)…copy this cell down through cell F75

REFERENCES

Pritzker, Yan. 2019. Inventing Bitcoin. Independently published.