CompuCoin Threat Model

In this document, we use ABC [6] to build a threat model for CompuCoin, the running example in ABC paper, which was inspired by Golem [1].

System description. CompuCoin is a cryptocurrency that provides a distributed computation outsourcing service. Parties with excessive CPU power may join the system as servers to perform computations on demand for others. Clients submit computation jobs to servers, wait for the results and proofs of correctness, and then pay these servers with cryptocurrency tokens for successfully completed jobs. This operation flow is captured by the network model shown in Figure 1. The mining process in CompuCoin is tied to the amount of service provided to the system. In other words, the probability of a server being selected to mine the next block on the blockchain is proportional to the amount of computation it has performed during a specific period of time.

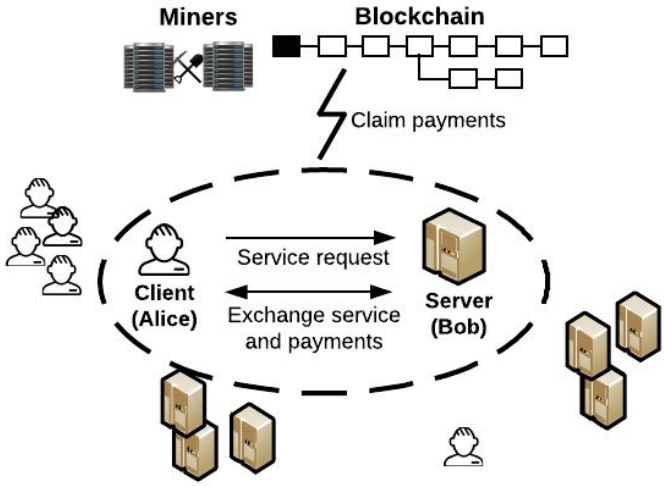

Figure 1: CompuCoin network model, computation outsourcing service.

Participants. Servers and clients (with servers filling the role of miners).

Dependencies. May rely on a verifiable computation outsourcing protocol, e.g., [2].

Assets. Service, service rewards ot payments, and the currency exchange medium assets, namely, the blockchain, transactions, the currency, and the communication network.

Assumptions. All messages exchanged in the system are signed using a secure digital signature scheme.

Threat categories identification. By mapping the assets in CompuCoin to the ABC threat categories (see Table 1 in [6]) and considering the system assumptions, we find that the following threats need to be investigated:

- **Service asset related threats:** Service corruption, denial of service, information disclosure.
- **Service payment asset related threats:** Service slacking and service theft.
- **Blockchain asset related threats:** Inconsistency, invalid blocks adoption, biased mining.
- **Transactions asset related threats:** Deanonymization.
- **Currency asset related threats:** Currency theft.
- **Communication network asset related threats:** Denial of service.

In CompuCoin, we assume that all messages exchanged are signed. Hence, repudiation and tampering are ruled out.

Threat scenarios enumeration and reduction. We construct 11 collusion matrices, one for each threat, and enumerate/reduce all possible threat scenarios. This involves crossing out the unlikely-to-happen threat cases and merging the ones that have identical effect.

We divide the service operation in CompuCoin into service sessions. During a service session a client interacts with a server to outsource computations as shown in Figure 1. In addition, the mining process is divided into rounds, where during a round a leader is elected to extend the blockchain with a new block.

In these matrices, the cells that are in black represent the ruled out cases while the cells in pink represent the merged ones. Inside these cells the rationale behind the omission or merging is outlined. Cells that contain right arrow and a comma-separated threat numbers indicate that colluding attackers do not become stronger than when acting individually. Each one may attack the system on its own and perform the attack(s) it is capable of from the comma-separated list.

We split the roles of the parties in the threat model. An external party, for example, can join the system as a server/client/miner and perform any of their activities. Same for server/clients/miners, they can perform any attack strategy an external is capable of. However, we do not repeat the same strategy for each one, but instead list it only once.

We remark that the enumerated set of attack strategies in each of these matrices could be extended in case we have a more detailed description of CompuCoin. However, we only rely on the brief description outlined in the paper as the purpose is to clarify the application of ABC steps.

1) Computation outsourcing service corruption threat

A service corruption threat in the context of CompuCoin means that clients receive invalid results of the computations they outsourced. This could be a result of any of the following actions:

- A server returns invalid results to the client deliberately.
- An attacker intercepts the communication between clients and servers and tampers the exchanged messages in a way that corrupts the results and/or corrupts the computation request, which makes the server perform different computation from the originally requested.

Note that the second strategy is ruled out due to the use of secure digital signature scheme to sign all messages exchanged in the system. Thus, we are left with the first strategy only as shown in Figure 2.

Based on the analysis of the collusion matrix shown in Figure 2, we have the following threat cases listed as tuples of threat type, attacker, and target:

- **Threat 1 (Computation outsourcing service corruption, attacker is server, target is client):** A server returns invalid computation results to a client deliberately.
    - The use of verifiable outsourced computation protocol allows detecting this attack, but does not prevent a server from practicing such behavior.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="3">Cannot be attackers, do not provide a service and cannot tamper/forge messages.</td><td rowspan="6">Cannot be a target, they do not receive a service, and cannot tamper/forge messages.</td><td rowspan="6">Reduced to the case of attacking client since a server cannot be a target.</td></tr>
<tr><td>Client</td></tr>
<tr><td>Client and External</td></tr>
<tr><td>Server</td><td>(1) Produce invalid computation results deliberately.</td></tr>
<tr><td>Server and Client</td><td rowspan="3">Reduced to the case of attacker server, colluding with other clients/external/other servers will not make the attacker stronger.</td></tr>
<tr><td>Server and External</td></tr>
<tr><td>Client, Server, and External</td></tr>
</table>

Figure 2: Collusion matrix of computation outsourcing service corruption threat.

2) Computation outsourcing DoS threat

DoS is a large threat category and covers several assets, e.g., blockchain, transactions, etc. In this section, we consider only the scenarios related to computation outsourcing service, which include the following:

- An attacker monitors the communication links and drops all service requests sent by a specific client(s), and/or all results sent back by a specific server(s).
- A server does not reply to service requests coming from a specific client(s).
- A client does not send service requests to a specific server(s).

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>Client</td><td>(2) Submit computation requests with large rewards so servers favor working with this client over others.</td><td>(3) Do not send computation requests to this miner.</td><td rowspan="3">Cannot do anything, parties already agreed on the service.</td></tr>
<tr><td>Server</td><td>(4) Do not respond to computation requests coming from this client.</td><td>(5) Ask for low service rewards so clients favor to work with this server over others.</td></tr>
<tr><td>Client and Server</td><td>→ (2), (4)</td><td>→ (3), (5)</td></tr>
<tr><td>External</td><td>(6) Drop all computation requests sent by clients.</td><td>(7) Drop all computation results sent by servers.</td><td rowspan="4">→ (7)</td></tr>
<tr><td>Client and External</td><td>→ (2), (6)</td><td>→ (3), (7)</td></tr>
<tr><td>Server and External</td><td>→ (4), (6)</td><td>→ (5), (7)</td></tr>
<tr><td>Client, Server, and External</td><td>→ (2), (4), (6)</td><td>→ (3), (5), (7)</td></tr>
</table>

Figure 3: Collusion matrix of computation outsourcing DoS threat.

The collusion matrix of this threat is found in Figure 3. In this matrix, the column header that has a client and server label means that these two parties have already agreed on the service. That is, the server accepted the client service request and the client is waiting the computation results. The analysis of this collusion matrix produces the following threat scenarios:

- **Threat 2 (Computation outsourcing DoS, attacker is client, target is client):** A client offers to pay high service rewards for the computation requests he submits, and thus, make servers favor his requests over others.
- **Threat 3 (Computation outsourcing DoS, attacker is client, target is server):** A client(s) do not send computation requests to specific server(s).

- **Threat 4 (Computation outsourcing DoS, attacker is server, target is client):** A server do not respond to computation outsourcing requests coming from a specific client(s).
- **Threat 5 (Computation outsourcing DoS, attacker is server, target is server):** A server sets the price of the service he provides to be low enough to make clients favor him over other servers.
- **Threat 6 and 7 (Computation outsourcing DoS, attacker is anyone, target is client/server):** An attacker drops all computation requests and results sent to the network. The attacker may attack a specific entity or all entities in the system to disrupt the service.

3) Computation outsourcing service information disclosure threat

The computation outsourcing service in CompuCoin involves exchanging computation requests and results. These information are sent in the clear (we do not assume the use of secure channels). As such, anyone can read the content of these messages. As such, the whole collusion matrix of this threat, which we omit, is reduced to the following threat scenario:

- **Threat 8 (Computation outsourcing service information disclosure, attacker is anyone, target is client/servers):** Anyone can read the content of all exchanged service requests and results, and hence, track the parties activities in the system.

4) Computation outsourcing service slacking threat

Recall that in this threat threat servers try to obtain payments with less work than promised. This does not include the case when a server outsources the computation to other servers. In other words, we are not concerned whether the server performed the computation locally or not as long as the promised work to a client is done as agreed. Instead, what we care about is that server provides partial or no results at all but still collects full payments from clients.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="3">Cannot be an attacker, it does not provide a service.</td><td rowspan="6">Cannot be a target, it does not pay for the service.</td><td rowspan="6">Reduced to the case of attacking a client.</td></tr>
<tr><td>Client</td></tr>
<tr><td>Client and External</td></tr>
<tr><td>Server</td><td rowspan="3">(9) Collect full payments without performing all the outsourced computations as promised.</td></tr>
<tr><td>Server and External</td></tr>
<tr><td>Server and Client</td></tr>
<tr><td>Client, Server, and</td><td></td><td></td></tr>
</table>

External			

Figure 4: Collusion matrix of computation outsourcing service slacking threat.

The collusion matrix of this threat is depicted in Figure 4, which shows the following threat case:

- **Threat 9 (Computation outsourcing service slacking, attacker is server, target is client):** A server performs the computation partially or even does not do any work at all but still collects full payments.

Note that the description of the threat is not concrete, this is the best we can say given the brief description of CompuCoin. In other systems, e.g. Filecoin, we show detailed strategies on how an attacker may pursue the service slacking threat.

5) Computation outsourcing service theft threat

Here, clients are trying to obtain full and valid service with less payments than promised. Given the brief description of CompuCoin, we have two scenarios for this threat as shown in the collusion matrix found in Figure 5 as follows:

- **Threat 10 (Computation outsourcing service theft, attacker is client, target is server):** A client does not issue payments to servers after obtaining a full and valid service as requested.
- **Threat 11 (Computation outsourcing service theft, attacker is client, target is server):** A client issues invalid payments to servers after obtaining a full and valid service as requested.

Other strategies, such as an attacker drops all payments issued by a client, or miners ignore adding these payments to the blockchain are part of the DoS attack against the communication network as will be discussed later.

Target → / **Attacker ↓**	***Client***	***Server***	***Client and Server***
External	**Cannot be a target, it does not serve others.**	**Cannot attack, it does not pay for the service.**	**Reduced to the case of attacking servers only, clients do not provide a service.**
Server			
Server and External			
Client		**(10)** Does not pay after obtaining the service. **(11)** Issue invalid payments.	
Client and External			

Server and Client		→ **(10), (11)**	
Client, Server, and External			

Figure 5: Collusion matrix of computation outsourcing service theft threat.

6) Blockchain inconsistency threat

We consider all strategies that may cause an inconsistency of the blockchain across miners either instantly or in the future. In other words, an attacker's actions may either make miners hold inconsistent copies of the blockchain now, or later on once the differing blocks are confirmed. These strategies are as follows:

1. An attacker makes miners process different versions of a transaction by performing double spending.
2. An attacker drops/withholds some transactions/blocks after being accepted by some miners.
3. An attacker controls the network connections of miners, i.e. control their view of the network, to make them build different versions of the blockchain.
4. An attacker forks the blockchain beyond the latest unconfirmed blocks and announces the new branches to different groups of miners.

Note that the aforementioned strategies do not involve tampering of transactions and blocks. Similar to Bitcoin, we assume all transactions are signed and that mining a block is done only by the selected miner has a proof of this selection that cannot be forged.

Recall that in CompuCoin the miners are the same as the servers that provide computation outsourcing service. In the discussion and the collusion matrices we alternate between the use of phrases servers and miners. Moreover, recall that the only target for all blockchain related threats are servers/miners because they are the parties that hold copies of the blockchain.

Analyzing the collusion matrix of this threat, as depicted in Figure 6, produces the following threat cases:

- **Threat 12 (Blockchain inconsistency, attacker is anyone, target is miner):** An attacker drops/withhold transactions and/or blocks causing miners to work on different branches of the blockchain.
- **Threat 13 (Blockchain inconsistency, attacker is anyone, target is miner):** An attacker controls the network view of the miners, i.e. what transactions/blocks they receive, by controlling their connectivity. Thus, miners work on different copies of the blockchain.
- **Threat 14 (Blockchain inconsistency, attacker is miner, target is miner):** A miner ignores blocks coming from other miners, and hence, bypass these blocks when extending the blockchain.

- **Threat 15 (Blockchain inconsistency, attacker is miner, target is miner):** A miner(s) may try to fork the blockchain as follows:
    a. Generate two (or more) different blocks when elected as the round leader. This is called nothing at the stake attack.
    b. Miners that were elected as leaders in the previous rounds collude with each other to create a new different branch of the blockchain by recreate different blocks and sign them.

We remark that the double spending strategy is discussed under the currency theft threat as will be shown shortly.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="7">Clients do not maintain the blockchain.</td><td>(12) drop/withhold/selectively relay transactions and/or blocks.
(13) control the connectivity of the miners/servers in the network.</td><td rowspan="7">Reduced to the case of attacking miners/servers.</td></tr>
<tr><td>Client</td><td>(20) double spending.</td></tr>
<tr><td>Client and External</td><td>→ (12), (13), (20)</td></tr>
<tr><td>Server</td><td>(14) ignore other miners/servers' blocks.
(15) fork the blockchain.</td></tr>
<tr><td>Server and External</td><td>→ (12), (13), (14), (15)</td></tr>
<tr><td>Server and Client</td><td>→ (14), (15), (20)</td></tr>
<tr><td>External, Client, and Server</td><td>→ (12) - (15), (20)</td></tr>
</table>

Figure 6: Blockchain inconsistency threat collusion matrix.

7) Invalid block adoption threat

The collusion matrix of this threat is depicted in Figure 7. Again, miners (or servers) are the only targets here because they are responsible of maintaining the blockchain. As shown, the matrix is reduced to the following threat case (as mentioned previously, double spending discussed under the currency theft threat):

- **Threat 16 (Invalid block adoption, attacker is miner, target is miner):** The elected leader miner includes invalid transactions in the its block, or mines on top of an invalid branch of the blockchain.

Note that the case of a malicious client issuing invalid transactions, on its own without colluding with miners, is ruled out. This is because under the assumption that the majority of the mining power is honest these transactions are not accepted. Although the phrase majority here may not be clear compared to the case of Bitcoin, for example, which depends on proof of work.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="7">Clients do not maintain the blockchain.</td><td>Cannot attack, honest miners will not accept invalid/tampered transactions and blocks.</td><td rowspan="7">Reduced to the case of miner/server as a target.</td></tr>
<tr><td>Client</td><td>(20) double spending.</td></tr>
<tr><td>Client and External</td><td>→ (20)</td></tr>
<tr><td>Server</td><td>(16) Mine/accept invalid blocks/transactions.</td></tr>
<tr><td>Server and External</td><td>→ (16)</td></tr>
<tr><td>Server and Client</td><td>→ (16), (20)
Here a client may issue invalid transactions as well that are accepted by colluding miners.</td></tr>
<tr><td>Server, Client, and External</td></tr>
</table>

Figure 7: Collusion matrix of invalid block adoption threat.

8) Biased mining threat

The election of the leader miner for each round is based on the amount of computation a server/miner contributes in the system. CompuCoin description does not provide details on how this selection is done. However, in general each server will be selected with a probability proportional to the amount of outsourced computation it performed. Servers may try to bias the leader election process by pretending to contribute large amount of outsourced computation service.

Figure 8 shows the collusion matrix of the biased mining threat, which outlines the following threat cases:

- **Threat 17 (Biased mining, attacker is miner, target is miner):** A miner pretends to perform a large amount of outsourced computation in the system to bias the leader election process.
- **Threat 18 (Biased mining, attacker is miner and client, target is miner):** A miner colludes with a client(s) to pretend that they were being served correctly.

Again, the above scenarios are not concrete due to lack of detailed description on how the service is handled in CompuCoin.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="7">Cannot be targets, they do not participate in mining.</td><td rowspan="2">Will not attack, attackers are not part of the mining process.</td><td rowspan="7">Reduced to the case of attacking servers only, clients do not participate in mining.</td></tr>
<tr><td>Client</td></tr>
<tr><td>Client and External</td><td rowspan="2">(17) pretend to perform large amount of outsourced computation.</td></tr>
<tr><td>Server</td></tr>
<tr><td>Server and External</td><td>→ (17)</td></tr>
<tr><td>Server and Client</td><td>(18) A client sends large number of computation outsourcing request either with or without being in need for the service.</td></tr>
<tr><td>External, Client, and Server</td><td>→ (17) and (18)</td></tr>
</table>

Figure 8: Biased mining threat collusion matrix.

9) Transaction deanonymization threat

We assume that in CompuCoin the blockchain is public, and hence, it is susceptible to this threat. Hence, the whole matrix of this threat, which we omit, is reduced to the following threat case:

- **Threat 19 (Transaction deanonymization, attacker is anyone, target is anyone inside the system):** An attacker is able to read the transaction content on the blockchain and could be able to compromise users anonymity and privacy.

10) Currency theft threat

An attacker may pursue a currency theft threat by performing any of the following strategies:

1. An attacker forges valid transactions that spend other's currency.
2. An attacker tampers transactions issued in the system to make itself the destination of the currency transfer.
3. A miner (server in our case) pretends to be the owner of a newly mined block to collect the mining rewards.
4. An attacker double spends its currency.
5. An attacker spoofs parties in the system so the source, voluntarily, sends funds to this attacker instead of the legitimate destination.

The first, second, and third strategies are ruled out by the security of the digital signature scheme used in signing all messages exchanged in the system. That is, forging signatures or tampering messages without invalidating the original signature succeed with negligible probability.

The collusion matrix of this threat is found in Figure 9. Analysing this matrix produces the following threat cases:

- **Threat 20 (Currency theft, attacker is client, target is anyone):** An attacker client double spends its currency when paying others.
- **Threat 21 (Currency theft, attacker is client, target is client):** An attacker spoofs other parties in the system to claim being the legitimate destination of a currency transfer.

<table>
<tr><th>Target →
Attacker ↡</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>Client</td><td>(20) double spending.
(21) spoofing other parties in the system to become the destination of fund transfer.</td><td colspan="2" rowspan="5">Reduced to the case of attacking a client. Miners are viewed as clients when they transact with their currency.</td></tr>
<tr><td>Client and server</td><td>→ (20), (21)
Note that by colluding with miners double spending becomes easier.</td></tr>
<tr><td>External</td><td rowspan="3">Reduced to the case of an attacker client, servers/externals are viewed as clients when they transact with their currency</td></tr>
<tr><td>Client and External</td></tr>
<tr><td>Server and External</td></tr>
<tr><td>Client, Server, and External</td><td>Reduced to the case of an attacker client colluding with miner/server.</td><td colspan="2"></td></tr>
</table>

Figure 9: Currency theft threat collusion matrix.

11) Communication network DoS threat

This threat involves all strategies that are part of the DoS against the computation outsourcing services, and these that are part of the blockchain inconsistency and chain freezing threats. In what follows, we cover the strategies that were not covered previously, which include:

- An external party takes the system down (i.e. Goldfinger attack).
- Issue huge number of transactions and service requests to overwhelm the network/miners.

Figure 10 shows the collusion matrix of this threat, where we have the following threat cases:

- **Threat 22 (Communication network DoS, attacker is anyone, target is the system network):** An attacker takes the whole network down.
- **Threat 23 (Communication network DoS, attacker is client/anyone, target is the system network):** An attacker overwhelm the network with huge number of service requests and/or transactions. This can be done by both issuing new requests/transaction and replaying old ones.

<table>
<tr><th>Target →
Attacker ↓</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td colspan="3">(22) take the system down.</td></tr>
<tr><td>Client</td><td colspan="3">(23) Overwhelm the network by issuing/replaying large number of requests and/or transactions.</td></tr>
<tr><td>Client and External</td><td colspan="3" rowspan="2">→ (22), (23)</td></tr>
<tr><td>Server</td></tr>
<tr><td>Server and External</td><td colspan="3" rowspan="3">Can perform any of the previous attacks based on the attackers combination.</td></tr>
<tr><td>Server and Client</td></tr>
<tr><td>External, Client, and Server</td></tr>
</table>

Figure 10: Communication network DoS threat collusion matrix.

References:

[1] Golem, https://golem.network/

[2] Parno, Bryan, Jon Howell, Craig Gentry, and Mariana Raykova. "Pinocchio: Nearly practical verifiable computation." In Security and Privacy (SP), 2013 IEEE Symposium on, pp. 238-252. IEEE, 2013.

[3] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In: Security and Privacy (SP), 2015 IEEE Symposium on. pp. 104–121. IEEE (2015).

[4] Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 281–310. Springer (2015).

[5] Pass, R., Seeman, L., Shelat, A.: Analysis of the blockchain protocol in asynchronous networks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 643–673. Springer (2017).

[6] Almashaqbeh, G., Bishop, A., Cappos, J.: ABC: A Cryptocurrency-Focused Threat Modeling Framework. In: The 2nd INFOCOM 2019 Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock), 2019.