# ABC: A Cryptocurrency-Focused Threat Modeling Framework

Ghada Almashaqbeh[1], **Allison Bishop**[1,2], Justin Cappos[3]

[1]Columbia, [2]Proof Trading, [3]NYU

**CryBlock 2019, Paris, France**

# Outline

- Background.
- Motivation.
- The ABC framework.
  - System model characterization.
  - Threat category identification.
  - Threat scenario enumeration and reduction.
  - Risk assessment and threat mitigation.
- User study.
- Use cases and experiences.
- Conclusion.

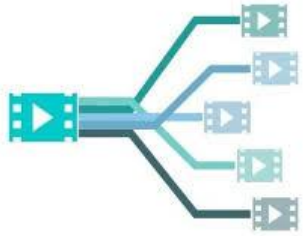
# Cryptocurrencies and Blockchain Technology

- An emerging economic force that received a huge interest.
- Started with Bitcoin in 2009.
  - Currently there are ***2135 cryptocurrencies****.
  - Total capital market exceeding $170 billion.
- Early systems focused on providing a virtual currency exchange medium.
  - Distributed; the trusted bank is replaced with the miners.
  - Publicly verifiable; everything is logged on the blockchain.
  - No real identities; anyone can join using a pseudonym.

*https://coinmarketcap.com/

# Cryptocurrency-based Distributed Services

- Provide distributed services on top of the currency exchange medium.
  - E.g., computation outsourcing (Golem), File storage (Filecoin), video transcoding (Livepeer).
- Any party can join to serve others in order to collect cryptocurrency tokens.
- The mining itself could be tied to the amount of service put in the system.
- Several economic aspects:
  - Could provide lower cost than centralized service providers.
  - A step forward on the “useful mining” path.



# But … Are They Secure?!

- Cryptocurrency/blockchain-based space experienced a huge number of attacks.
  - Financial incentives lead to more motivated attackers.
- Security is more challenging in cryptocurrency-based distributed services.
  - Complicated functionality.
  - Larger scale.
  - Usually open access model, anyone can join with no pre-identification.
  - Fair service-payment exchange is impossible between distrusted parties.

# Threat Modeling and Cryptocurrencies

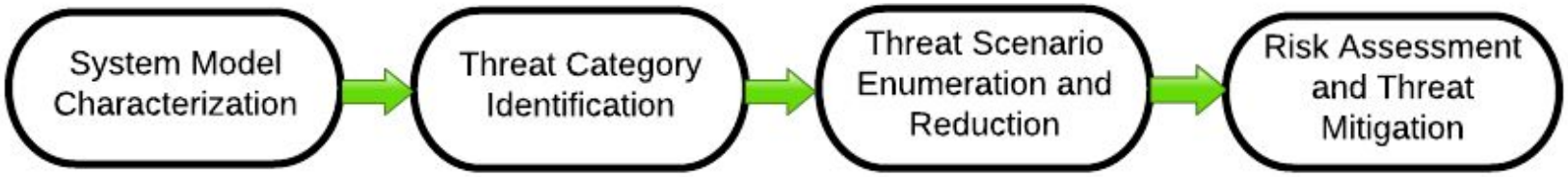
- Threat modeling is an essential step in secure systems design.
  - Explore the threat space to a system and identify the potential attack scenarios.
  - Helps in both guiding the system design, and evaluating the security of developed systems.
- Traditional approaches do not fit cryptocurrency-based systems.
  - Do not scale.
  - Do not explicitly account for attacker financial motivation nor collusion between these attackers.
  - Do not consider the new threat types cryptocurrencies introduce.

# ABC: Asset-Based Cryptocurrency-focused Threat Modeling Framework

# What is ABC?

- A systematic threat modeling framework geared toward cryptocurrency-based systems.
  - Its tools are useful for any distributed system.
- Helps designers to focus on:
  - Financial motivation of attackers.
  - New asset types in cryptocurrencies.
  - Deriving system-specific threat categories.
  - Spotting collusion and managing the complexity of the threat space.
    - Using a new tool called a collusion matrix.
- Integrates with other steps of a system design; risk management and threat mitigation.

# ABC Steps




Running Example: ***CompuCoin***

- A cryptocurrency that provides a distributed computation outsourcing service.
- Parties with excessive CPU power may join as servers to perform computations for others in exchange for CompuCoin tokens.
- The mining process is tied to the amount of service these servers provide.

# Step 1: System Model Characterization

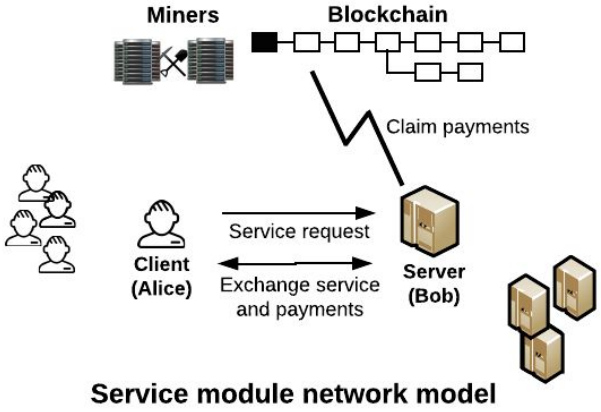
- Identify the following:
  - Activities in the system.
  - Participant roles.
  - Assets.
  - Any external dependencies on other services.
  - System assumptions.
- Draw a network diagram(s) of the system modules.

**Functionality description.** Outlined in CompuCoin description introduced earlier.

**Participants.** Clients and servers.

**Dependencies.** May rely on a verifiable computation outsourcing protocol.

**Assets.** Computation service, service rewards (or payments), blockchain, currency, transactions, and the communication network.




Service module network model

# Step 2: Threat Category Identification

- Define broad threat classes that must be investigated.
- ABC defines these classes around the assets.
- For each asset, do the following:
  - Define what constitute a secure behaviour for the asset.
  - Use that knowledge to derive the asset security requirements.
  - Define threat classes as violations of these requirements.

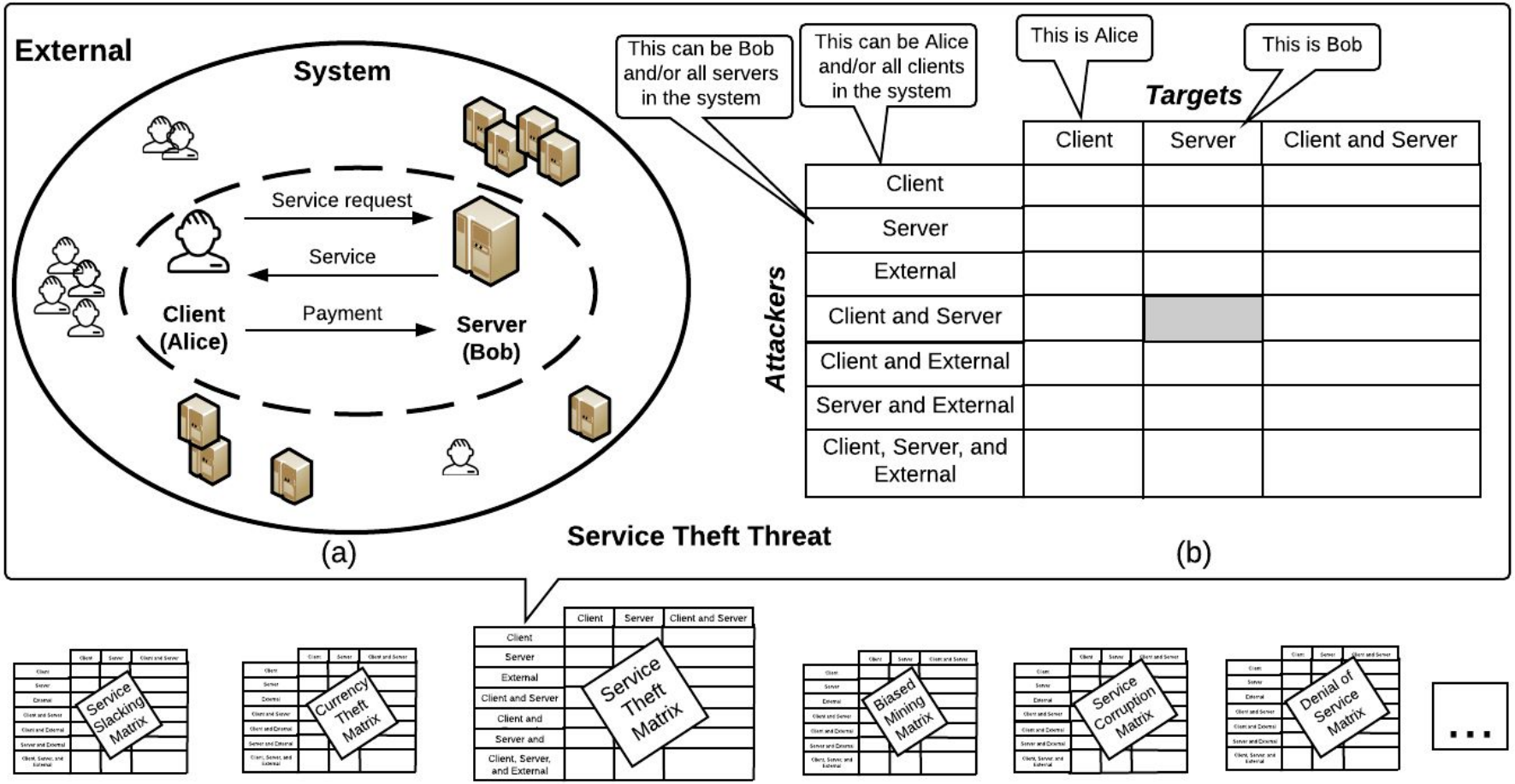
# Step 2: Running Example Application

| Asset | Security Threat Category |
|---|---|
| Service | Service corruption (provide corrupted service for clients). |
| | Denial of service (make the service unavailable to legitimate users). |
| | Information disclosure (service content/related data are public). |
| | Repudiation (the server can deny a service it delivered). |
| Service payments | Service slacking (a server collects payments without performing all the promised work). |
| | Service theft (a client obtains correct service for a lower payment than the agreed upon amount). |
| Blockchain | Inconsistency (honest miners hold copies of the blockchain that may differ beyond the unconfirmed blocks). |
| | Invalid blocks adoption (the blockchain contains invalid blocks that does not follow the system specifications). |
| | Biased mining (a miner pretends to expend the needed resources for mining to be elected to extend the blockchain). |
| Transactions | Repudiation (an attacker denies issuing transactions). |
| | Tampering (an attacker manipulates the transactions in the system). |
| | Deanonymization (an attacker exploits transaction linkability and violates users' anonymity). |
| Currency | Currency theft (an attacker steals currency from others in the system). |
| Communication network | Denial of service (interrupt the operation of the underlying network). |

# Step 3: Threat Scenario Enumeration and Reduction

- For each threat, define scenarios that attackers may follow to pursue their goals.
  - Be comprehensive, consider collusion and financial motivation.
- ABC devises collusion matrices to help with this step.
- Analyzing a collusion matrix involves:
  - Enumerating all possible attack scenarios.
  - Crossing out irrelevant cases and merge together those that have the same effect.
  - Documenting all distilled threat scenarios.

# Collusion Matrix



# Step 2: Running Example Application

## Service Theft Threat Collusion Matrix

<table>
<tr><th>Target<br>Attacker</th><th>Client</th><th>Server</th><th>Client and Server</th></tr>
<tr><td>External</td><td rowspan="7">Clients cannot be targets because they do not serve others.</td><td>Servers and external cannot attack because they do not ask/pay for service.</td><td rowspan="7">Reduced to the case of attacking servers only, clients do not serve others (cannot be targets).</td></tr>
<tr><td>Server</td></tr>
<tr><td>Server and External</td></tr>
<tr><td>Client</td><td>(1) Refuse to pay after receiving the service.<br>(2) Issue invalid payments.</td></tr>
<tr><td>Client and External</td><td rowspan="3">Reduced to the case of an attacker client. A client does not become stronger when colluding with other servers or external entities.</td></tr>
<tr><td>Server and Client</td></tr>
<tr><td>Client, Server, and External</td></tr>
</table>

# Step 4: Risk Management and Threat Mitigation

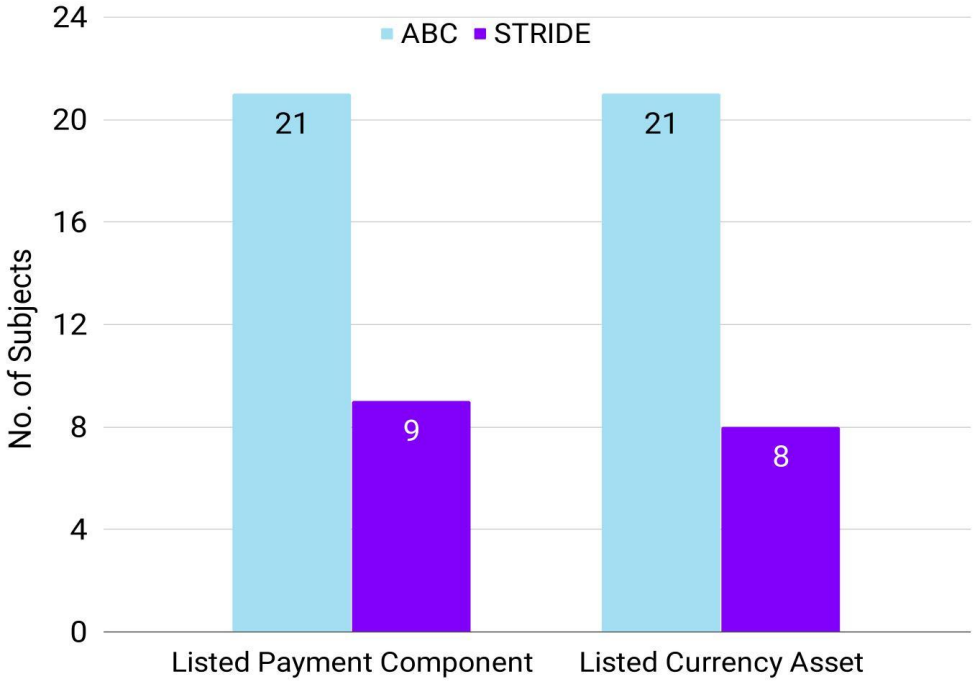
- An independent task of threat modeling.
- However, financial incentives affect prioritizing threats and their mitigation techniques.
  - Use game theory-based analysis to quantify the pay-off an attacker may obtain.
  - Use detect-and-punish techniques to address certain threat types.
- For example, in CompuCoin:
  - Locking payments in an escrow neutralizes threat 1.
  - Having a penalty deposit that is fortified upon cheating addresses threat 2.
  - Both require careful design and economic analysis.

# Evaluation - User Study

# User Study - ABC vs. STRIDE

- Recruited 53 participants (mainly security masters students).
  - 5 pilot run, two groups of 24 subjects (one tested STRIDE, one tested ABC).
- Asked to build a threat model for a cryptocurrency-based file storage and retrieval network called ArchiveCoin.
- Each session spanned 3 hours.
  - Overview of cryptocurrencies.
  - A tutorial for ABC or STRIDE.
  - Overview of ArchiveCoin.
  - Threat model building.

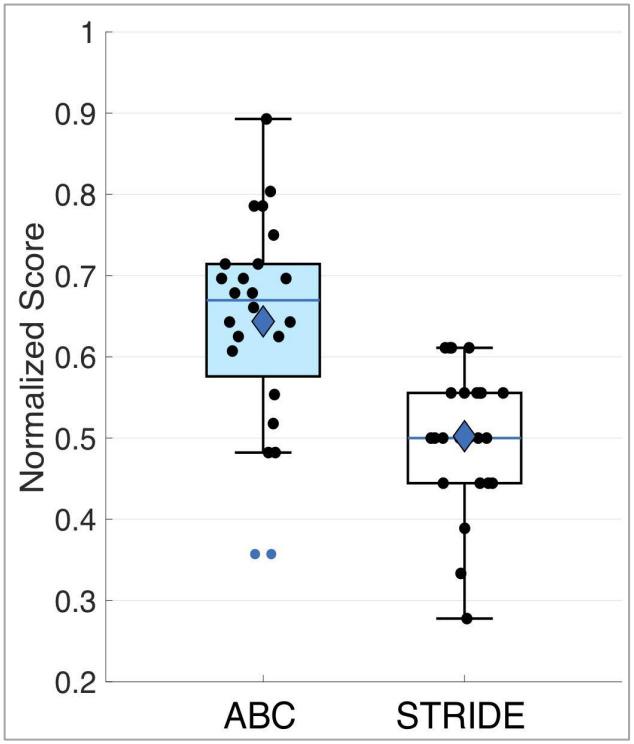
# Results - Financial Aspects and Collusion




- For financial threat in question (service theft of file retrieval):
  - STRIDE 13%, ABC 71%.
- For collusion: none in STRIDE, while 45% in ABC.

# Results - Accuracy

- Computed precision, recall, and total score.
  - Precision -- STRIDE 0.48, ABC 0.57
  - Recall -- STRIDE 0.4, ABC 0.48
  - Total scores (normalized).
    - STRIDE avg 0.5, ABC avg 0.64



# Evaluation - Use Cases

# Use Cases

- Applied ABC to three real world systems.
  - Bitcoin - well established system.
  - Filecoin - close to launch.
  - CacheCash - our system, under development.

- We developed ABC while working on CacheCash when we realized that none of traditional frameworks suited our needs.

# Use Cases - Outcome

| Aspect | Bitcoin | Filecoin | CacheCash |
|---|---|---|---|
| ABC steps covered | Steps 1-3 | Seps 1-3 | Steps 1-4 |
| Completion time (hr) | 10 | 47 | Not tracked |
| No. of collusion matrices | 5 | 14 | 9 |
| Threat cases total | 105 | 882 | 525 |
| Distilled threat cases | 10 | 35 | 22 |

- All known threats to Bitcoin were mapped to the collusion matrices ABC produced.
- Revealed ***3*** *unaddressed threats* in the public design of Filecoin.
- ABC was useful for CacheCash in both pre-design threat modeling step, and after-design security analysis.

# Extended Version

https://arxiv.org/abs/1903.03422

(full user study results and deeper discussion of the use cases)

# Conclusions

- Cryptocurrencies provide a disruptive work model.
  - But also exhibit complicated relations between, financially motivated, untrusted parties.
- Great potential and huge arena of applications.
  - However, deeper thinking is needed to assess when/where to apply.
  - Threat modeling is a critical step to enhance their security.
- Are they just a hype that will fade away?!
  - Still provide an elegant proof of concept.

# Questions?

aNd ThANk yOU :)