CSE5095-010: Blockchain Technology

Lecture 14

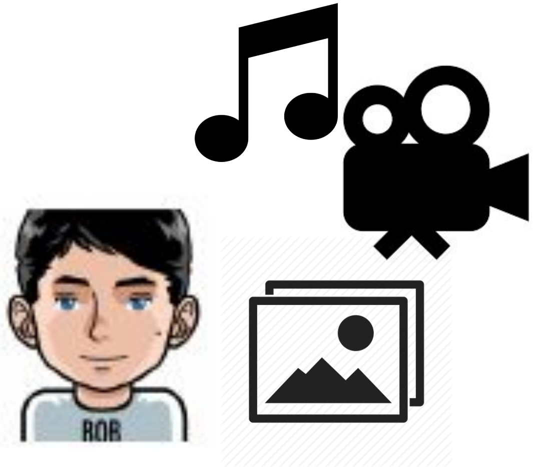
Ghada Almashaqbeh

UConn - Fall 2020

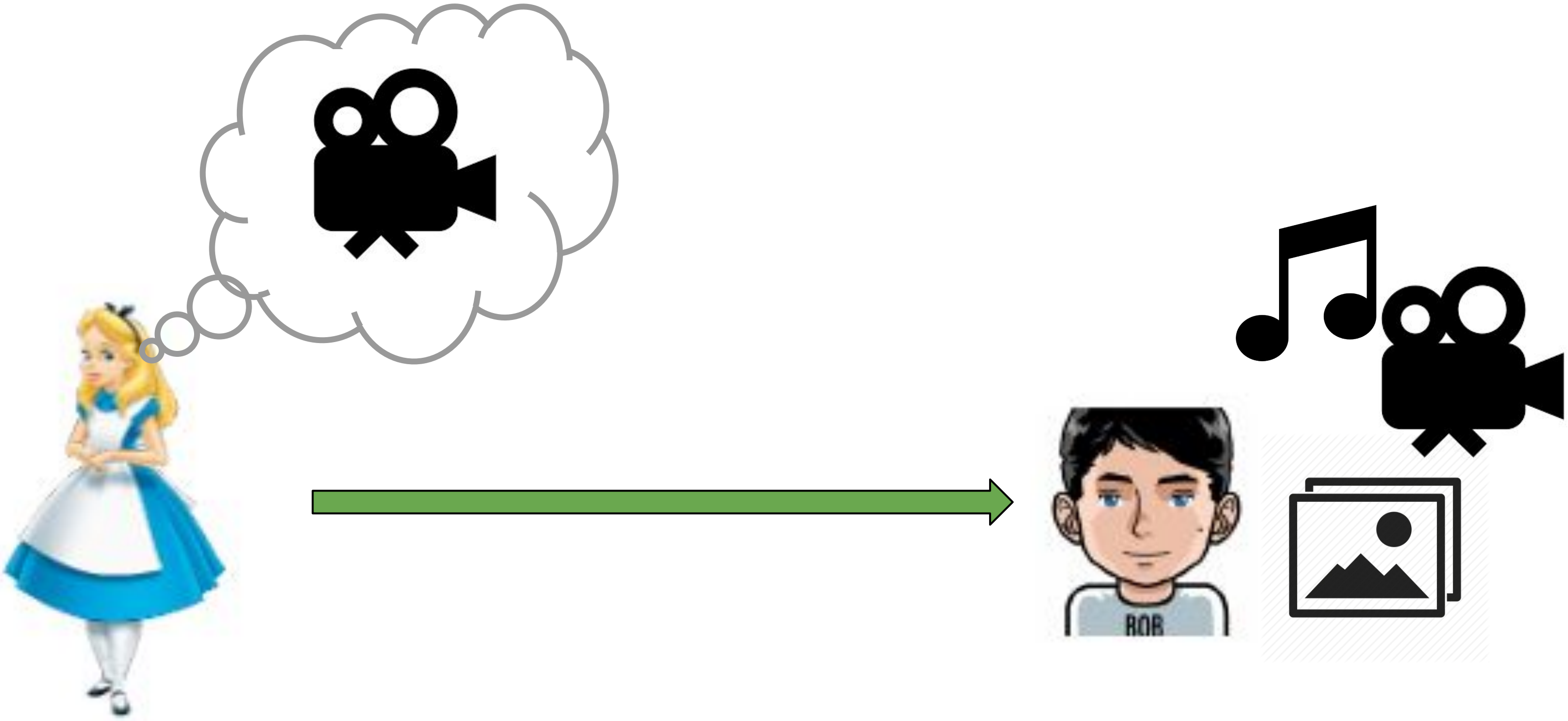
Outline

- Micropayments.
    - Motivation.
    - On the use of payment channels/networks for micropayments.
    - Probabilistic micropayments.
        - Centralized schemes.
        - Decentralized schemes.
            - MICROPAY.
            - DAM.
            - MicroCash.

Customer

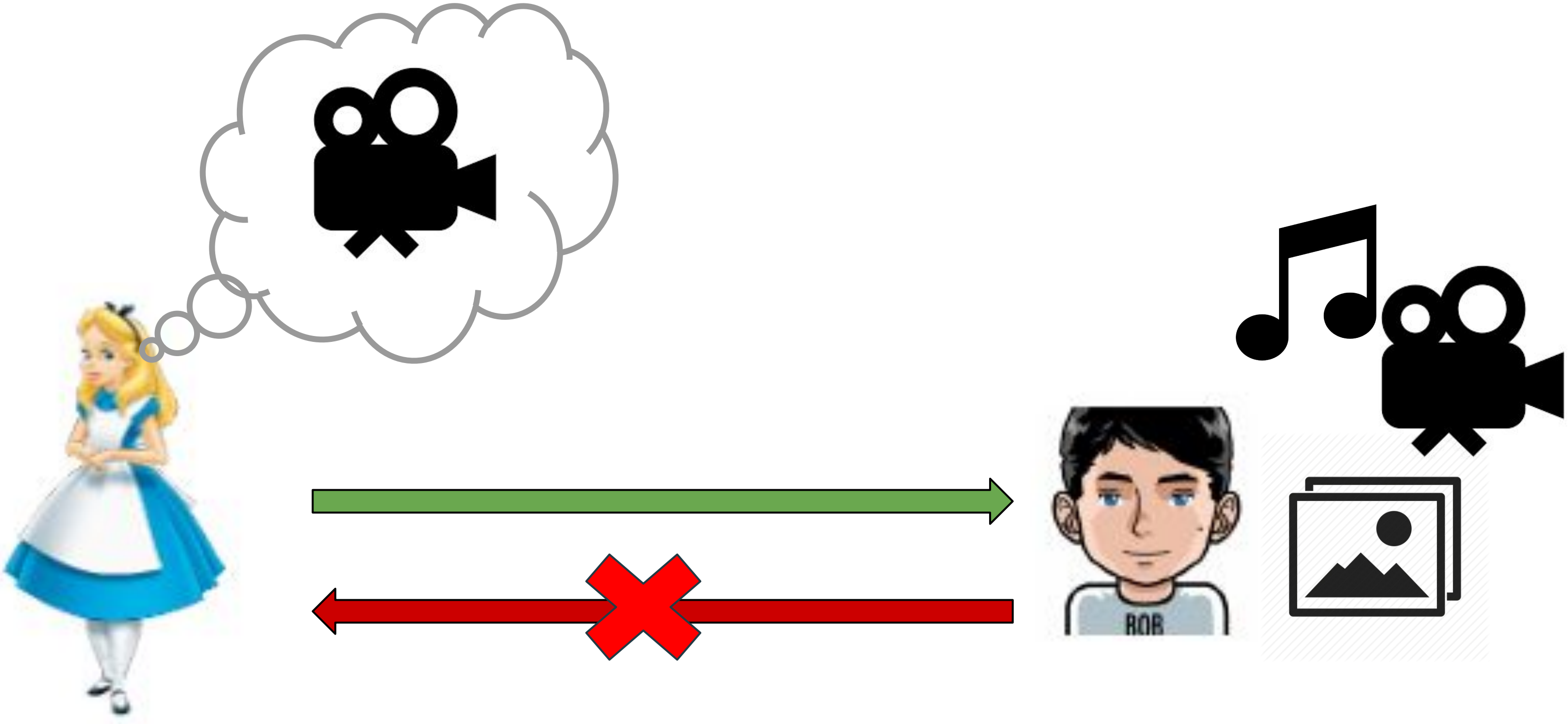


Merchant



Customer

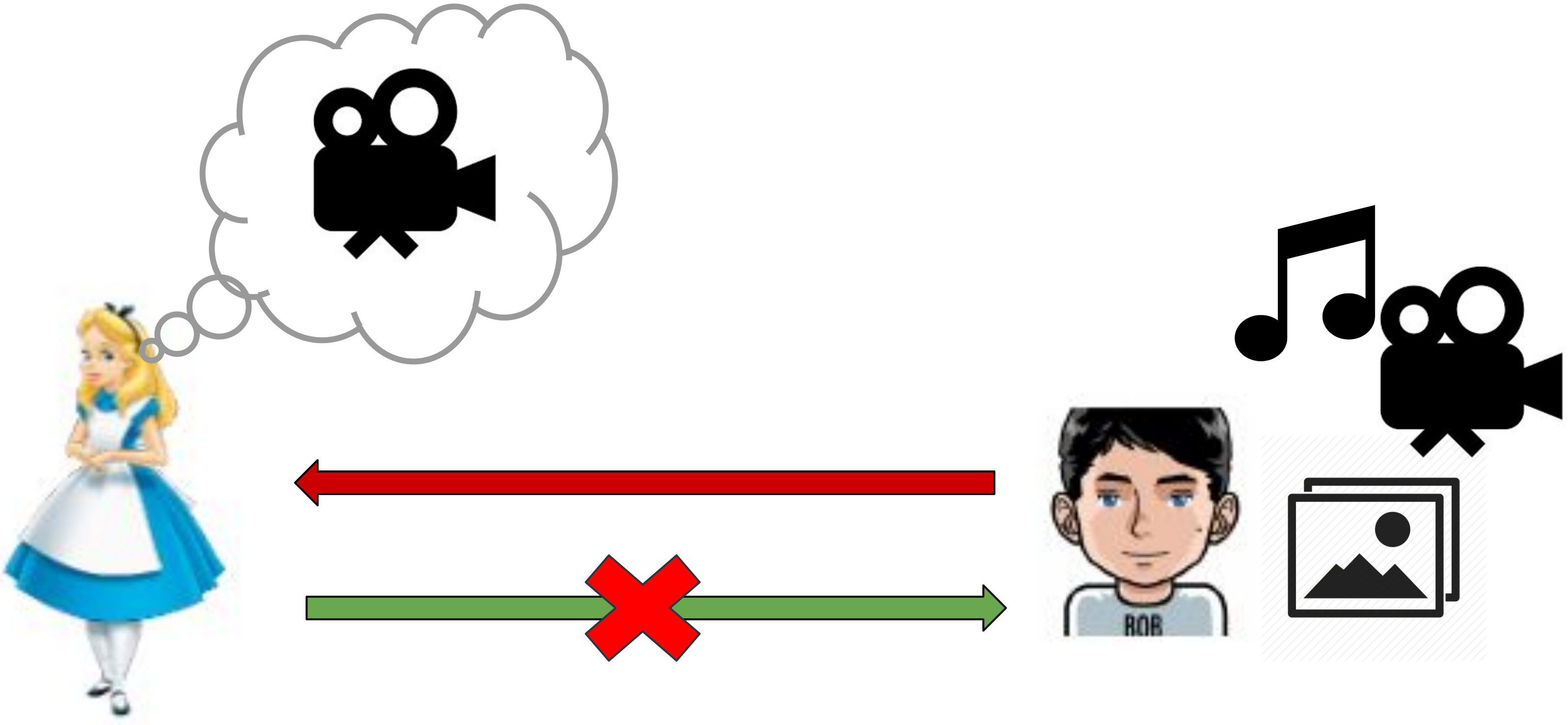
Merchant



Customer

Merchant

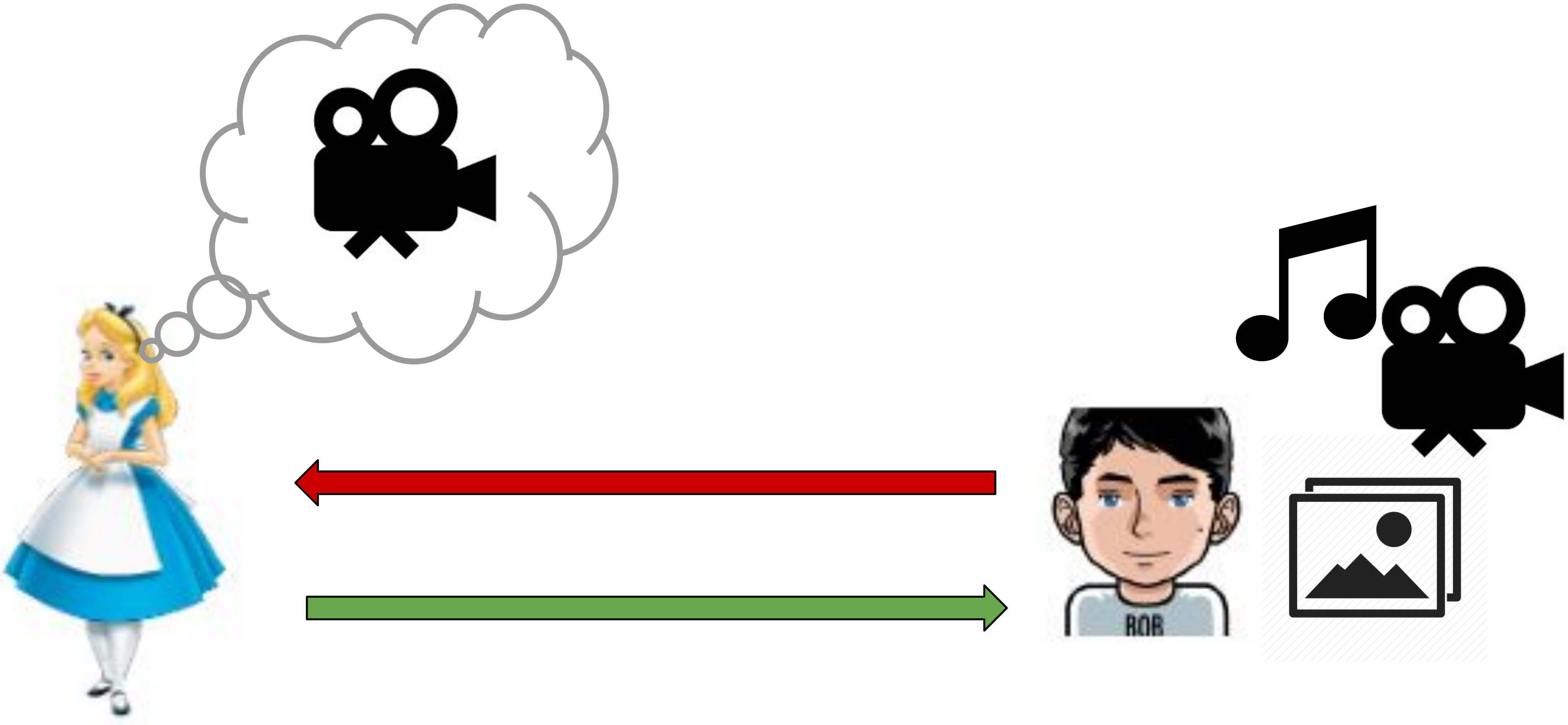
The Merchant could fail to provide the service and keep the customer's money



Customer

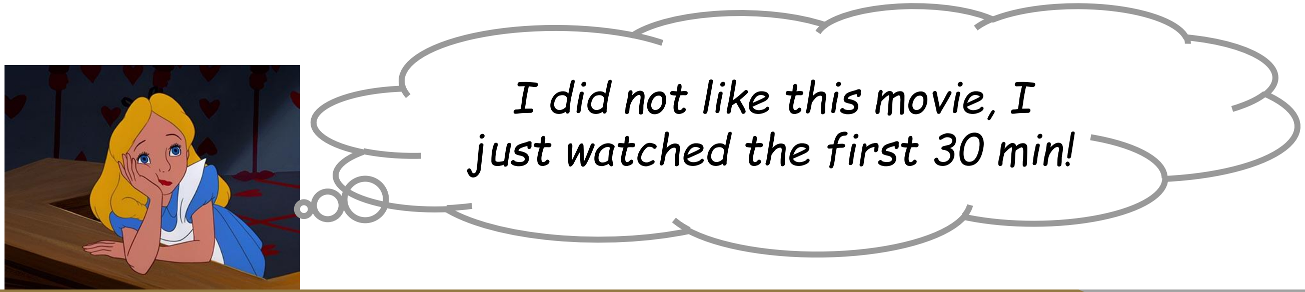
Merchant

The Customer could fail to pay after the merchant has provided the service



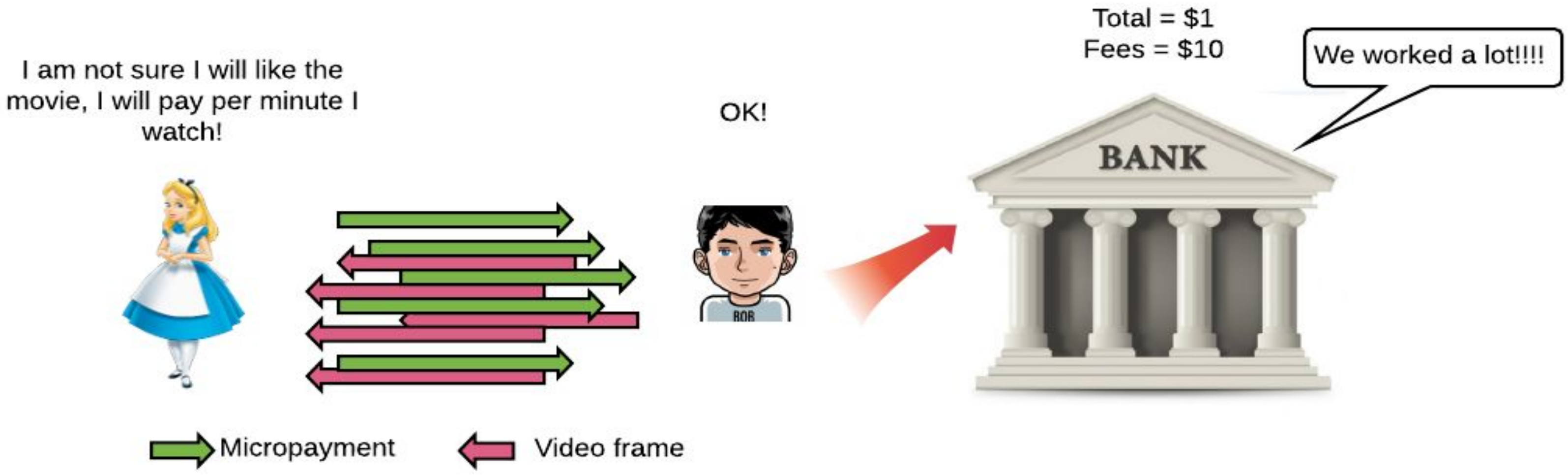
Customer

Merchant



I did not like this movie, I just watched the first 30 min!

Micropayments



- A payment of a micro value, i.e., pennies or fractions of pennies.
- Several applications, e.g., ad-free web, online gaming, etc.
  - Used extensively in cryptocurrency-based P2P distributed services.
  - Main motivation is the impossibility of fair service-payment exchange.

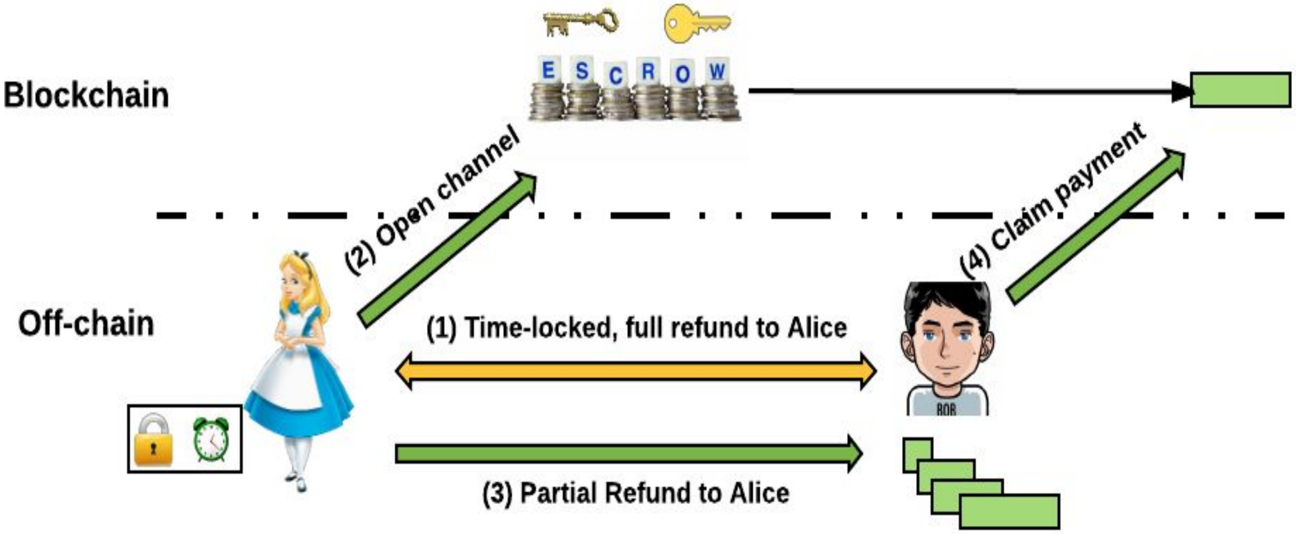
Challenges

- Produce a huge number of small-value transactions.
    - Overwhelm the system.
    - Explode the payment log.
    - Cannot scale for large demands or large number of users.
    - High transaction fees.
        - Each transaction must pay a fee.
        - This fee may exceed the payment value itself.

Aggregate the small payments into few larger ones!

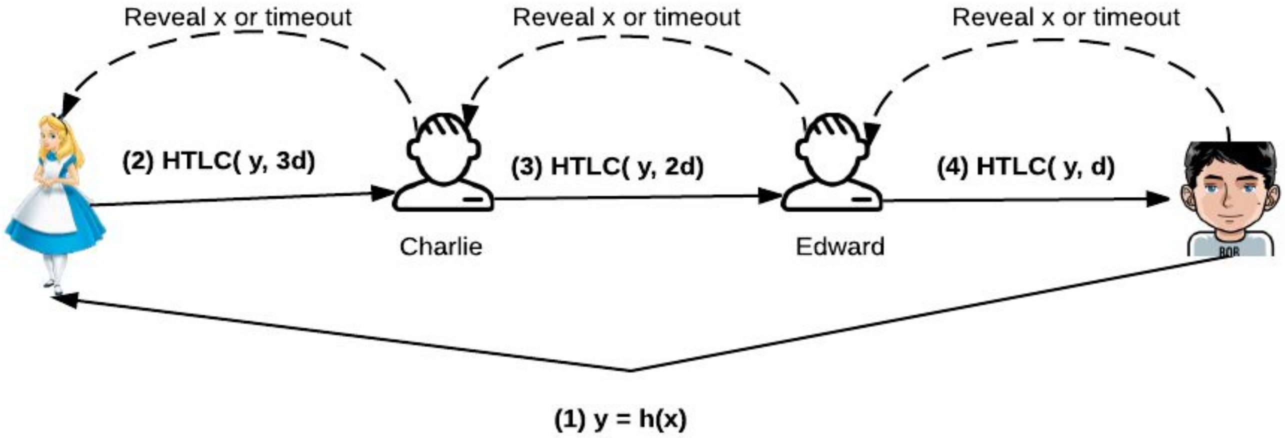
Micropayment Channels

- Process most transactions off-chain, only channel opening and closing transactions will be on-chain.
- A channel allows exchanging payments between only two parties.



Micropayment Networks

- Payment networks allow paying several parties.
  - E.g., the lightning networks.
  - Alice can pay Bob as long as there is a payment path between them.
    - Principal component: HTLC (Hash Time-Lock Contract).

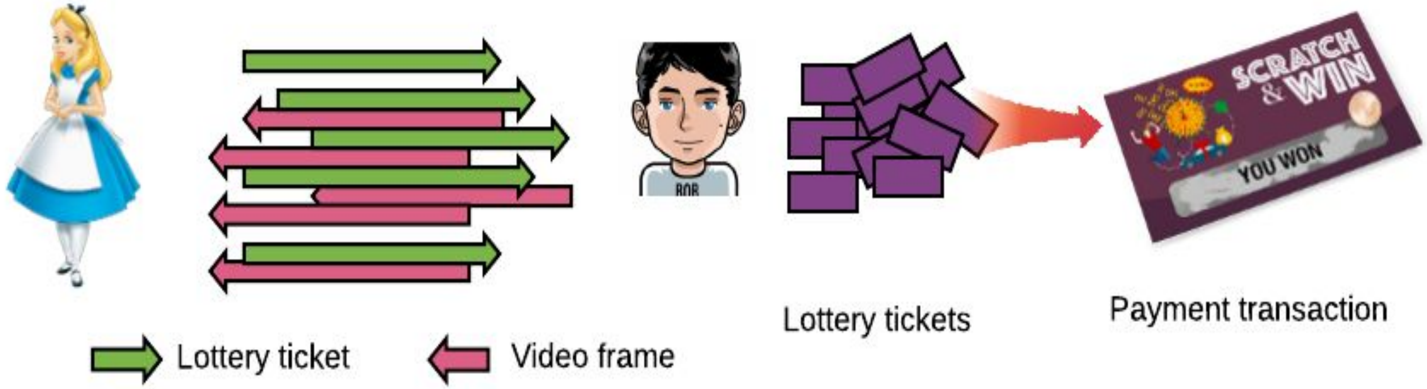


Issues

- Drive the system toward centralization.
  - Only wealthy parties can afford to be payment hubs.
- Hubs charge fees for relaying payments.
  - ***Fees are back!*** They may exceed the micropayment value itself.
- But, payment channels between long-term transacting parties (two parties) are still useful to handle micropayments.
- Currently payment networks are more geared towards enhancing scalability (i.e., transaction throughput rate) of cryptocurrencies.

Probabilistic Micropayments

- A solution to aggregate tiny payments.
- Dated back to Rivest [Rivest, 1997] and Wheeler [Wheeler, 1996].



Centralized Probabilistic Micropayments

- Early schemes were centralized.
- Involve a ***trusted bank*** to:
  - Authenticate users.
  - Hold users’ accounts.
  - Authorize customers to issue lottery tickets.
  - Audit the lottery and manage payments.
- We will explore the scheme of [Rivest, 1997].
  - The original version that is based on an interactive coin tossing protocol.

Rivest’s Scheme - Setup

- Beside creating accounts with the bank, the customer and merchant do the following:
  - The customer creates a hash chain

    $$x_0, x_1, x_2, \ldots, x_n, \text{ where } x_i = H(x_{i+1}).$$

  - The merchant creates a hash chain

    $$y_0, y_1, y_2, \ldots, y_n, \text{ where } y_i = H(y_{i+1}).$$

  - The merchant sends the root y_0 (signed) to the customer.
  - The customer sends the root x_0 concatenated with y_0 (signed) to the merchant.
    - This commits both parties to the hash chains they created.

Rivest’s Schemes - Payments

- A customer pays a merchant at round i by sending him x_i.
- A micropayment wins if ***x_i mod n = y_i mod n***
    - Where n = 1/p (must be an integer).
- Upon winning, the merchant sends the committed chain roots, in addition to x_i and y_i, to the bank.
    - The bank verifies that the ticket is a winning, valid one.
        - The validity of the chain, the lottery outcome, signatures, etc.
    - Then it transfers currency from the customer’s account to the merchant’s account.

Drawbacks - Centralization!

- Increases the deployment cost.
  - Establish relationships/accounts with bank.
- Limit the use of the payment service to systems with fully authenticated users.
- Drive the system toward centralization (trust and transparency issues!).
  - Not fully decentralized anymore.

Decentralized Probabilistic Micropayments

- Utilize blockchain/cryptocurrencies to convert centralized schemes into distributed ones.
- Ingredients:
  - The bank is replaced with the miners.
  - Escrows are created on the blockchain.
  - Consensus rules to manage escrows, claim/verify winning tickets, and punish cheaters.
- Three systems are out there:
  - **MICROPAY** [Pass et al., 2015],
  - **DAM** [Chiesa et al., 2017],
  - and **MicroCash** [Almashaqbeh et al., 2020].

MICROPAY1 [Pass et al., 2015] - Setup

- The customer creates an escrow with value X/p.
  - X is the expected value of a micropayment, and X/p is the value of a winning lottery ticket (i.e., total payment value).
  - This escrow can pay ***only one*** winning lottery ticket.
  - The escrow has its own public-private keypair.
    - The customer ***knows*** the private key of the escrow.
- So simply the customer creates a transaction transferring money to the escrow’s address.

MICROPAY1 - Payment

- The merchant asks for a payment (or a lottery ticket) as follows:
    - Select a random number r1,
    - Generate a commitment to r1 called c (like c = hash(r1)).
    - Generate a public key pkM.
    - Send (c, pkM) signed to the customer.
- The customer replies as follows:
    - Select another random number r2,
    - Send (r2, c, pkM) signed using the escrow private key back to the merchant.
- So it is a two-round (interactive) lottery protocol.

MICROPAY1 - Lottery

- A ticket wins if:

  r1 XOR r2 has log(1/p) leading zero digits

  (think about the XOR result in decimal).

- The merchant sends the lottery ticket (c, r1, r2, signature) to the miners.
  - This constitutes an unlocking script (in Bitcoin terms) to spend the escrow transaction.

MICROPAY1 - Issues

- Several issues:
  - ***Sequential*** ticket issuance under the same escrow.
  - ***Double spending:*** issue the same ticket to several merchants.
  - ***Front running:*** withdraw the escrow before a merchant claims its payment.
    - Both are mitigated financially by having a penalty escrow.
    - However, the amount of this penalty is ***not specified***.
  - ***Interactive lottery.***
    - A non-interactive lottery was introduced but it is computationally heavy.
  - Chances of having ***all tickets win*** (psychological obstacle to use the system).

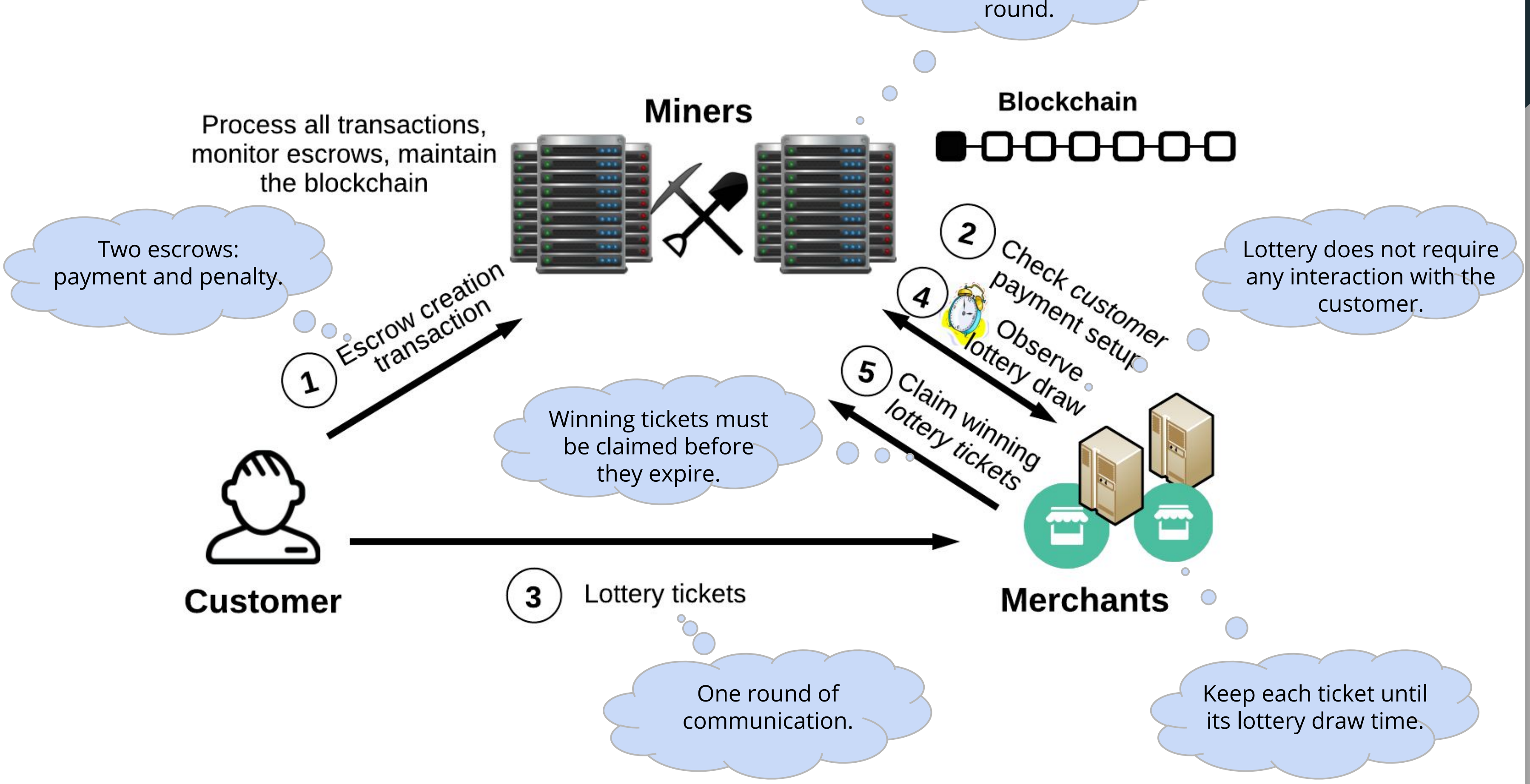
DAM [Chiesa et al., 2017]

- Addresses anonymity.
  - Built as an extension to ZeroCash.
- Solves:
  - Double spending: financially with a lower bound for the penalty deposit.
  - Front running: by delaying escrow withdrawal transactions.
- Issues:
  - Sequential.
  - Interactive lottery protocol.
  - Possibility that all tickets may win.
  - Computationally heavy.
    - For the additional machinery to support privacy/anonymity.

MicroCash [Almashaqbeh et al., 2020]

- The ***first*** decentralized probabilistic micropayment scheme that supports **concurrent micropayments**.
- The ***first*** to introduce a lottery with ***exact win rate.***
  - Non-interactive lottery requiring only secure hashing.
- Compared to sequential micropayment schemes, it reduces the amount of data on the blockchain by around **50%.**
  - This is due to the fact an escrow can pay multiple winning tickets.

MicroCash in a Nutshell


Miners
Blockchain
Process all transactions, monitor escrows, maintain the blockchain
1 Escrow creation transaction
2 Check customer payment setup
4 Observe lottery draw
5 Claim winning lottery tickets
3 Lottery tickets
Customer
Merchants

Produce lottery draw number for each round.

Two escrows: payment and penalty.

Lottery does not require any interaction with the customer.

Winning tickets must be claimed before they expire.

One round of communication.

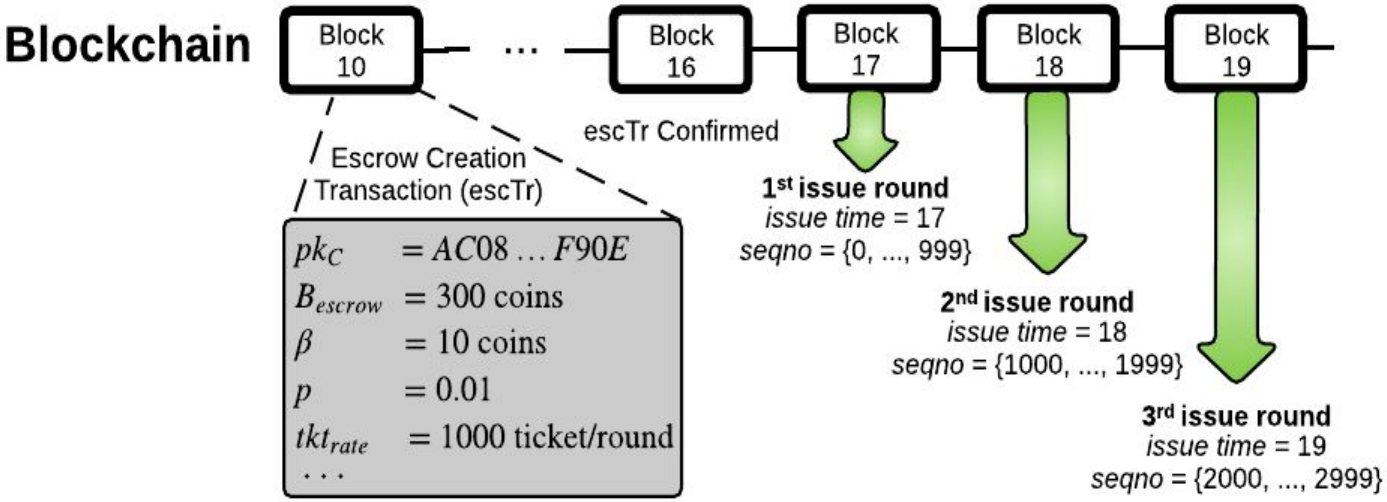
Keep each ticket until its lottery draw time.

Lottery Ticket Issuance

- Each ticket is a simple structure consist of:

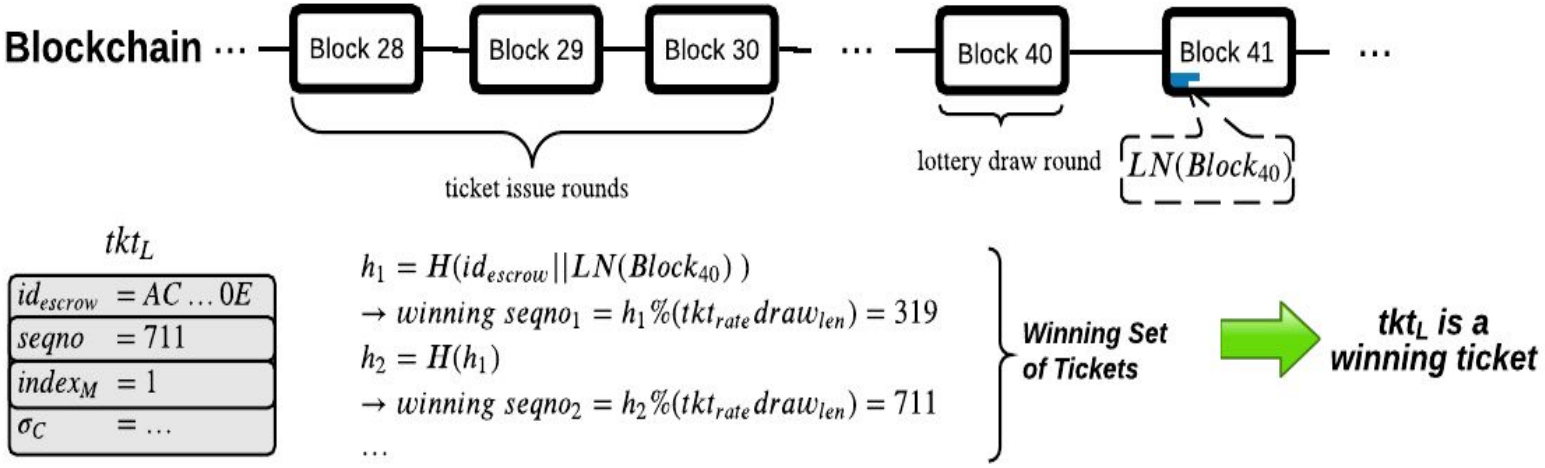
$$tkt_L = id_{esc}||index_M||seqno||\sigma_C$$

- Ticket issuance must follow a ticket issuing schedule.



The lottery Protocol

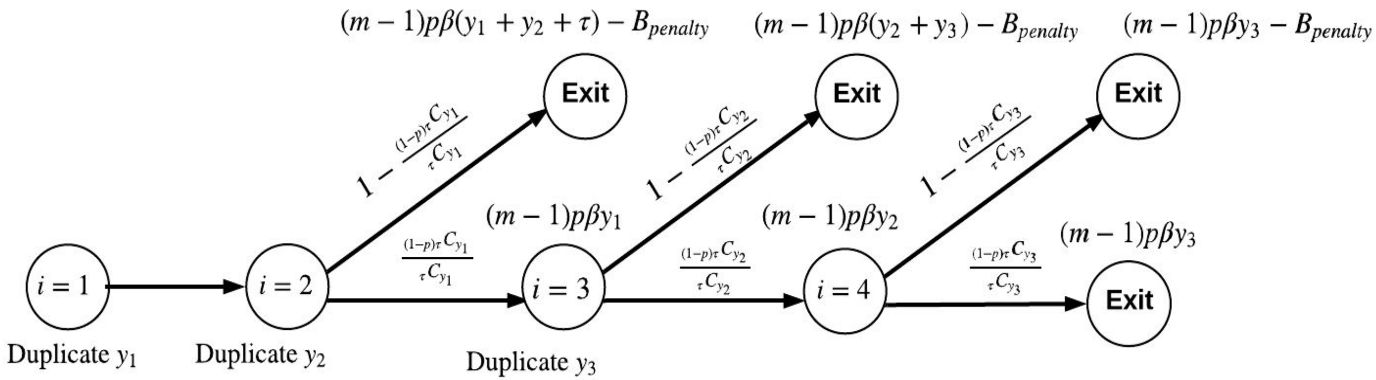
- Lightweight, non-interactive, and supports exact win rate.
  - Based on the blockchain view and requires only secure hashing.



$$
\begin{aligned}
h_1 &= H(id_{escrow}||LN(Block_{40})\,) \\
\rightarrow \textit{winning seqno}_1 &= h_1\%(tkt_{rate}draw_{len}) = 319 \\
h_2 &= H(h_1) \\
\rightarrow \textit{winning seqno}_2 &= h_2\%(tkt_{rate}draw_{len}) = 711 \\
&\dots
\end{aligned}
$$

Penalty Escrow

- Used to defend against ticket duplication.
  - Equals at least the additional utility a malicious customer obtains over an honest.



$(m-1)p\beta(y_1+y_2+\tau) - B_{penalty}$
$(m-1)p\beta(y_2+y_3) - B_{penalty}$
$(m-1)p\beta y_3 - B_{penalty}$
Exit
Exit
Exit
$1-\frac{{}_{(1-p)\tau}C_{y_1}}{{}_{\tau}C_{y_1}}$
$1-\frac{{}_{(1-p)\tau}C_{y_2}}{{}_{\tau}C_{y_2}}$
$1-\frac{{}_{(1-p)\tau}C_{y_3}}{{}_{\tau}C_{y_3}}$
$(m-1)p\beta y_1$
$(m-1)p\beta y_2$
$(m-1)p\beta y_3$
$\frac{{}_{(1-p)\tau}C_{y_1}}{{}_{\tau}C_{y_1}}$
$\frac{{}_{(1-p)\tau}C_{y_2}}{{}_{\tau}C_{y_2}}$
$\frac{{}_{(1-p)\tau}C_{y_3}}{{}_{\tau}C_{y_3}}$
$i=1$
$i=2$
$i=3$
$i=4$
Exit
Duplicate y_1
Duplicate y_2
Duplicate y_3

Theorem. *For the game setup of MicroCash, issuing invalid or duplicated lottery tickets is less profitable in expectation than acting in an honest way if:*

$$B_{penalty} > (m-1)p\beta\tau\left(\frac{1-p}{1-\frac{1}{{}_{\tau}C_{(1-p)\tau}}} + (1-p)(d-1) + r\right)$$

MicroCash - Issues

- Not fully compatible with any of the cryptocurrencies out there.
- To address double spending (and similar to DAM), the set of merchants that can be paid by using an escrow must be set in advance.
- Works in the random oracle model.

References

- [Rivest, 1997] Ronald Rivest.1997.Electronic lottery tickets as micropayments. In International Conference on Financial Cryptography. Springer, 307–314.
- [MR, 2002] Micali, Silvio, and Ronald L. Rivest. "Micropayments revisited." In Cryptographers’ Track at the RSA Conference, pp. 149-163. Springer, Berlin, Heidelberg, 2002.
- [Wheeler, 1996] David Wheeler. 1996. Transactions using bets. In International Workshop on Security Protocols. Springer, 89–92.
- [Pass et al., 2015] Pass, Rafael and Abhi Shelat. "Micropayments for decentralized currencies." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 207-218. ACM, 2015.
- [Chiesa et al., 2017] Alessandro Chiesa et al. "Decentralized Anonymous Micropayments." In EuroCrypt, 2017.
- [Almashaqbeh et al., 2020] Ghada Almashaqbeh et al. "MicroCash: Practical Concurrent Processing of Micropayments." In Financial Cryptography, 2020.