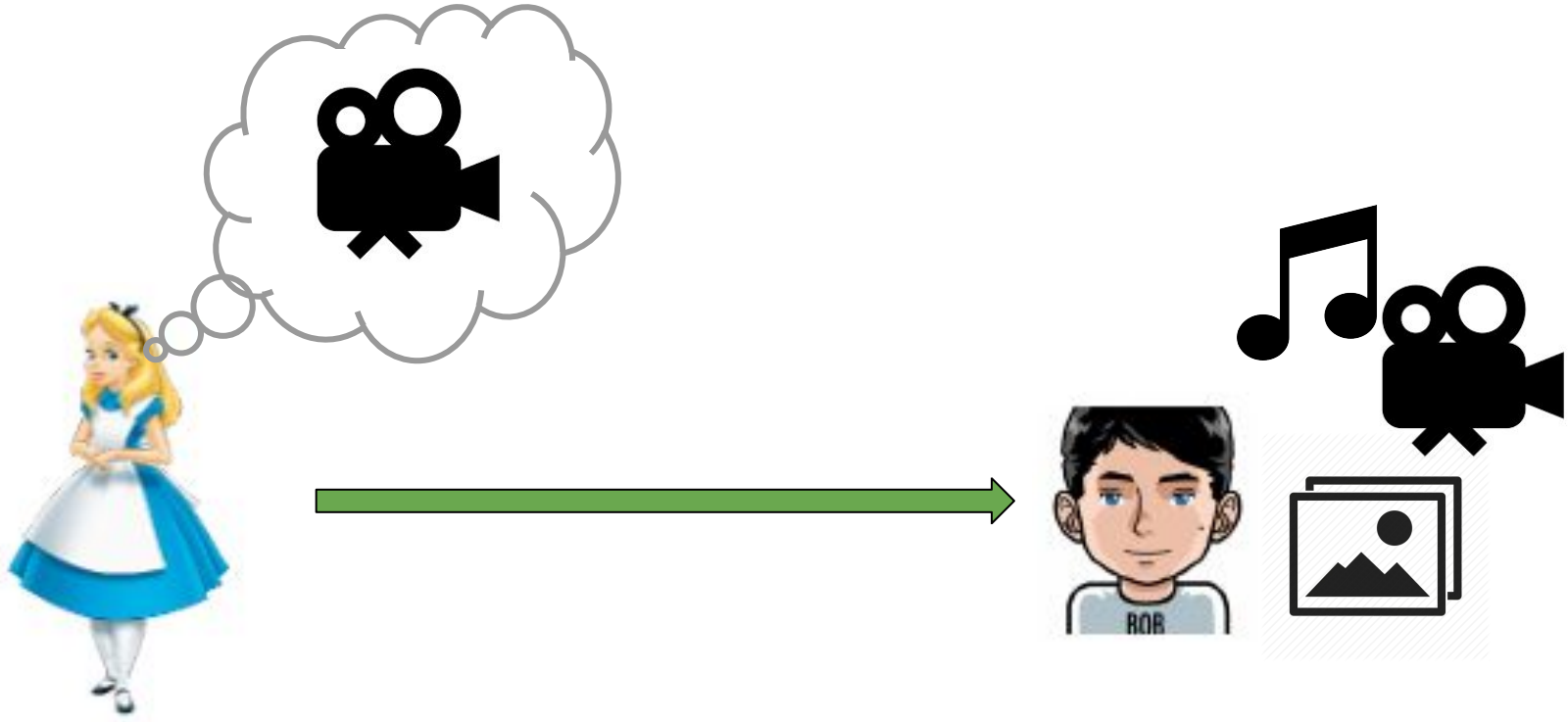
Micropayments

From Centralized to Blockchain-based Distributed Schemes

Ghada Almashaqbeh
NuCypher

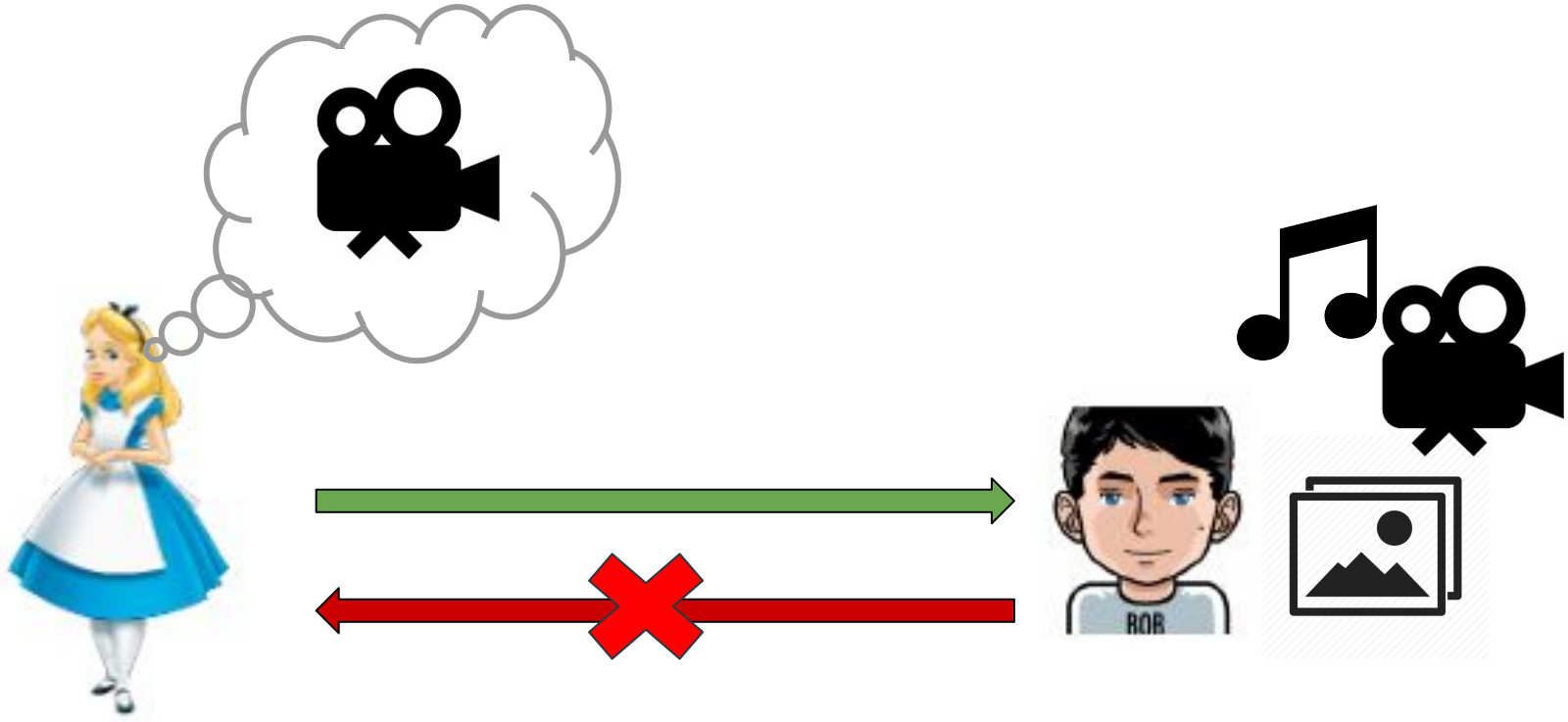
Customer

Merchant



Customer

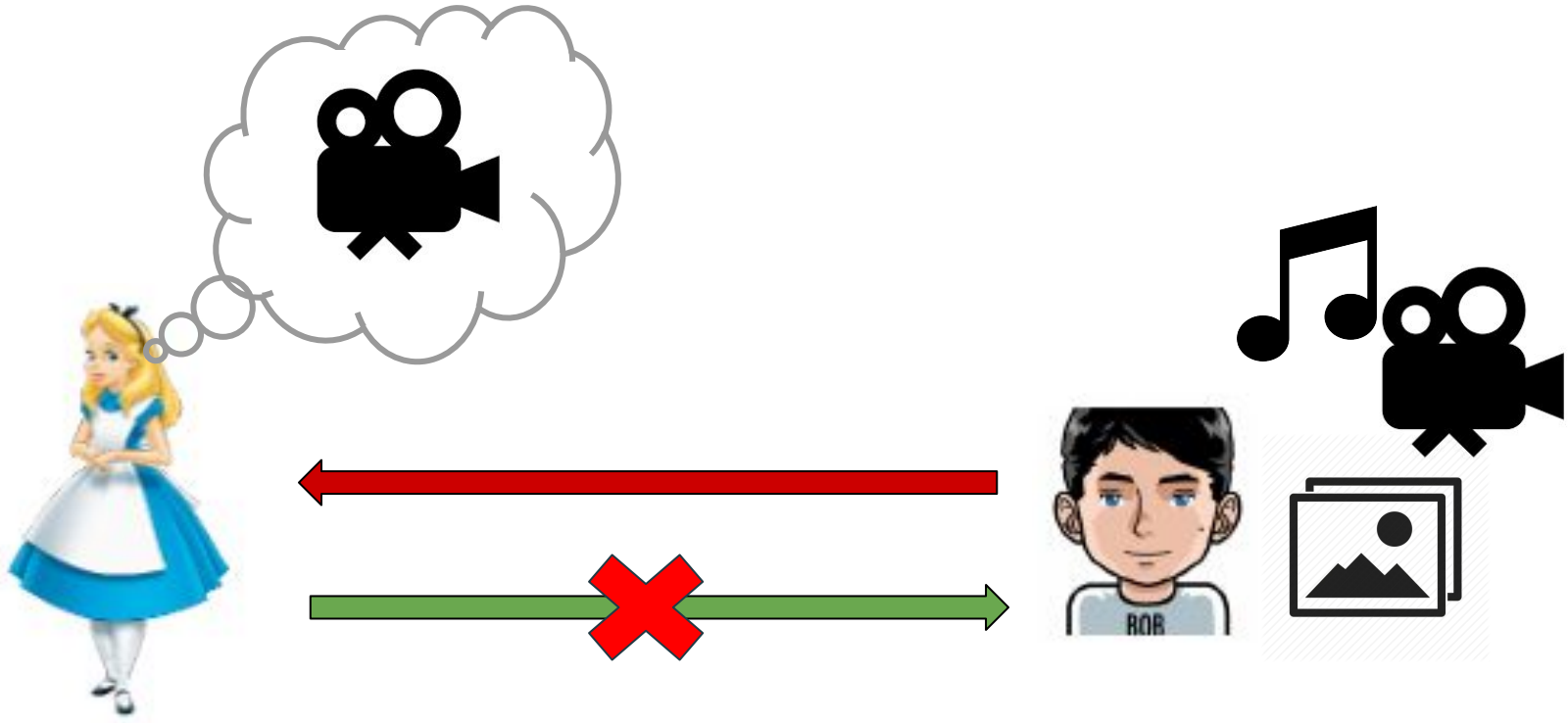
Merchant



Customer

Merchant

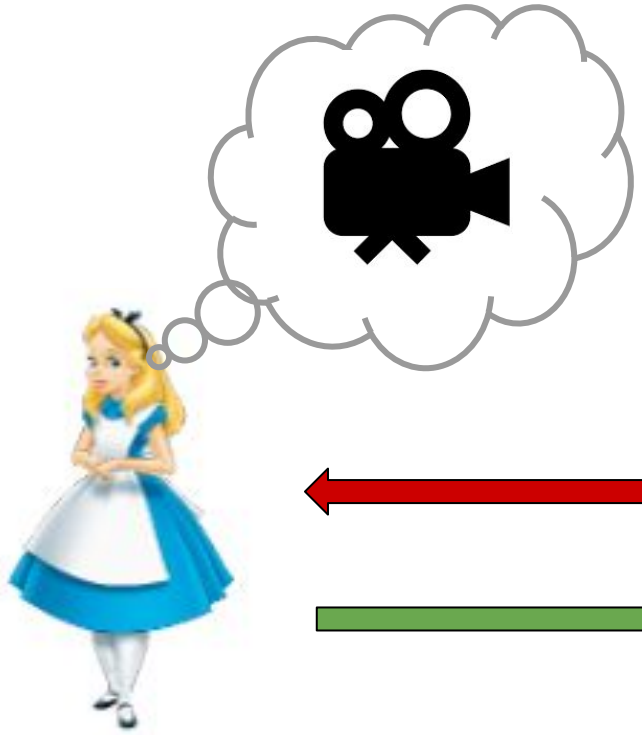
The Merchant could fail to provide the service and keep the customer's money

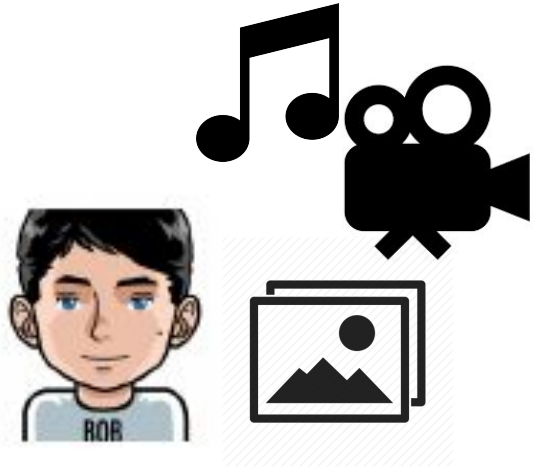


Customer

Merchant

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



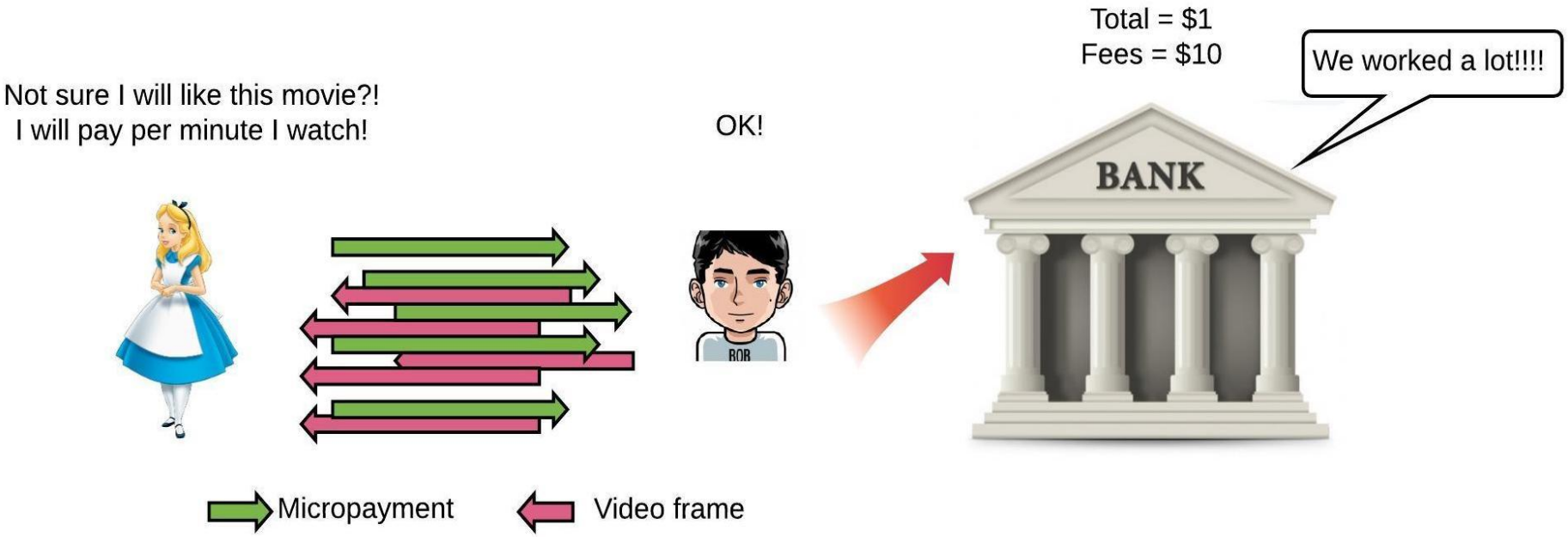


Customer

Merchant

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

Micropayments

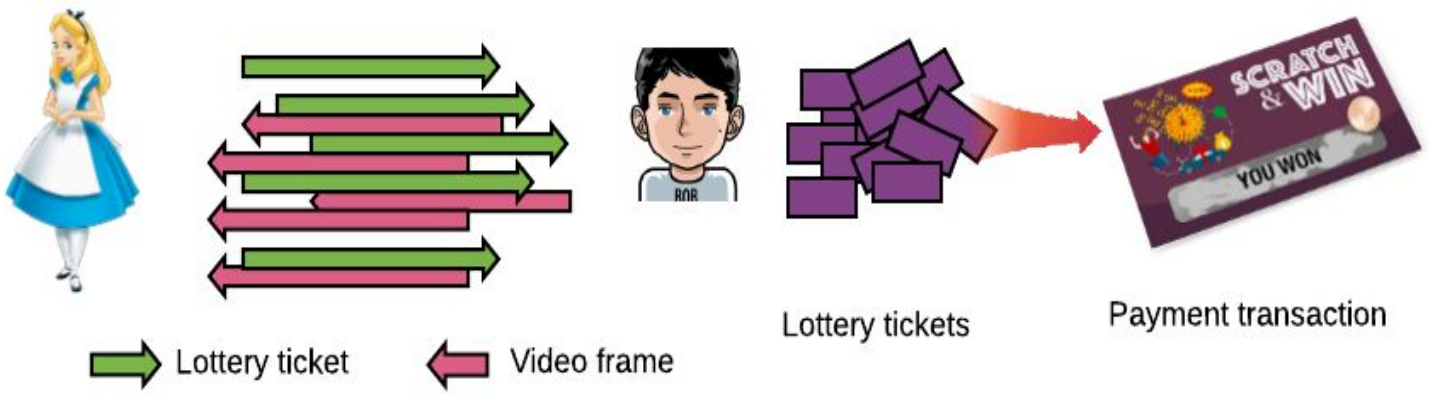


- A payment of micro value.
- Several applications, e.g., ad-free web, online gaming, etc.
- Suffer from high transactions fees and large payment log size.

Aggregate the small payments into few larger ones!

Probabilistic Micropayments

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



Centralized Probabilistic Micropayments

- 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 chain that each one created.

Rivest's Schemes - Payments

- A customer pays a merchant at round i by sending him x_i.
- A micropayment wins if $x_i \bmod n = y_i \bmod n$
  - 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 one.
  - 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 system with fully authenticated users.
- Drive the system toward centralization (trust and transparency issues!).

Decentralized Probabilistic Micropayments

- Utilize blockchain/cryptocurrencies to convert centralized schemes into distributed ones.
- Ingredients:
  - Replacing the bank with the miners.
  - Creating escrows on the blockchain.
  - Consensus rules to 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 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).

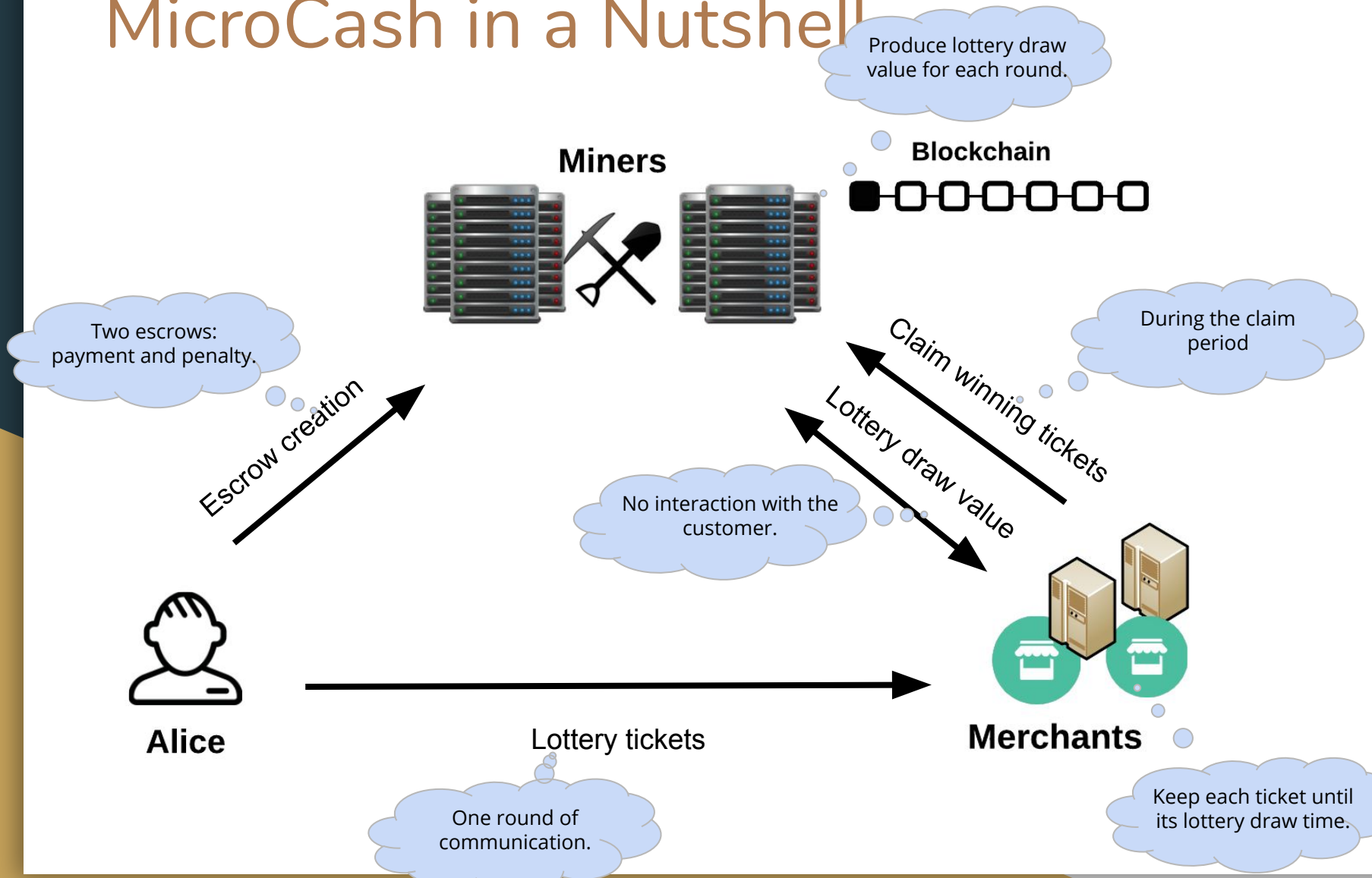
DAM [Chiesa et al., 2017]

- Addresses anonymity.
  - Built as an extension to Zcash.
- 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.

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.
- Reduces the amount of data on the blockchain by around **50%.**
  - compared to sequential micropayment schemes.
- Increases ticket processing rate by **1.7 - 4.2x**
  - compared to MICROPAY.

MicroCash in a Nutshell

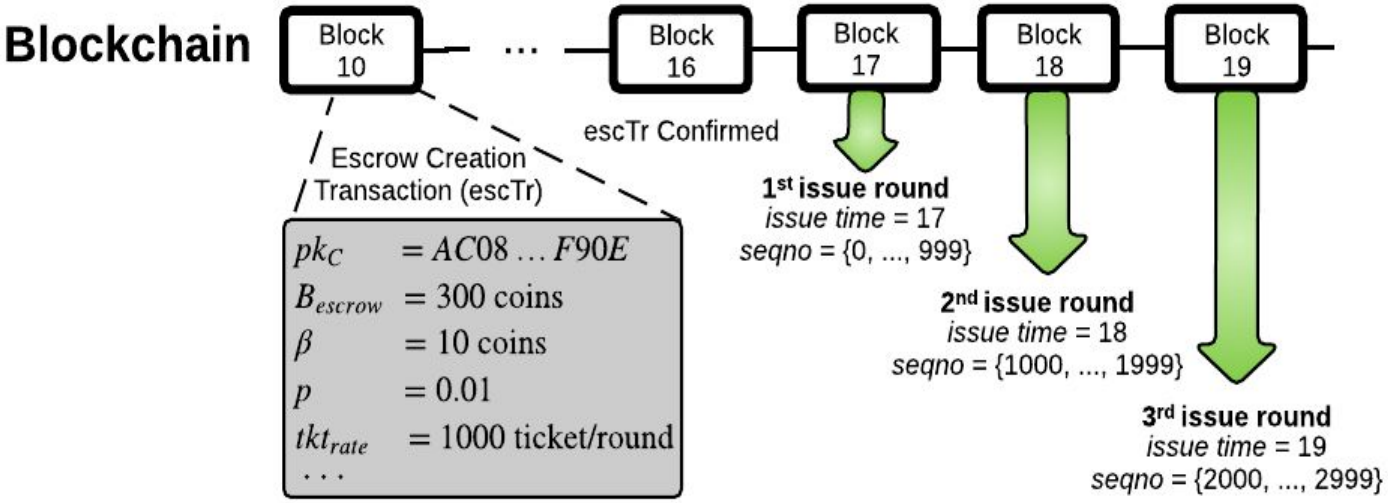


Lottery Ticket Issuance

- Each ticket is a simple structure consist of:

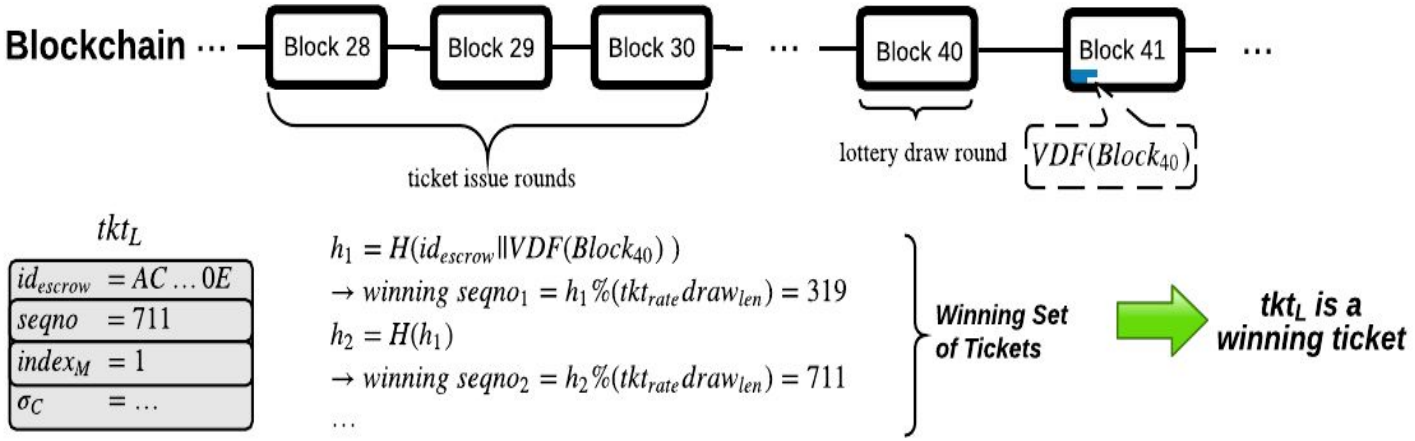
$$tkt_L = id_{esc} || index_M || seqno || \sigma_C$$

- Ticket issuance must follow a ticket issuing schedule.



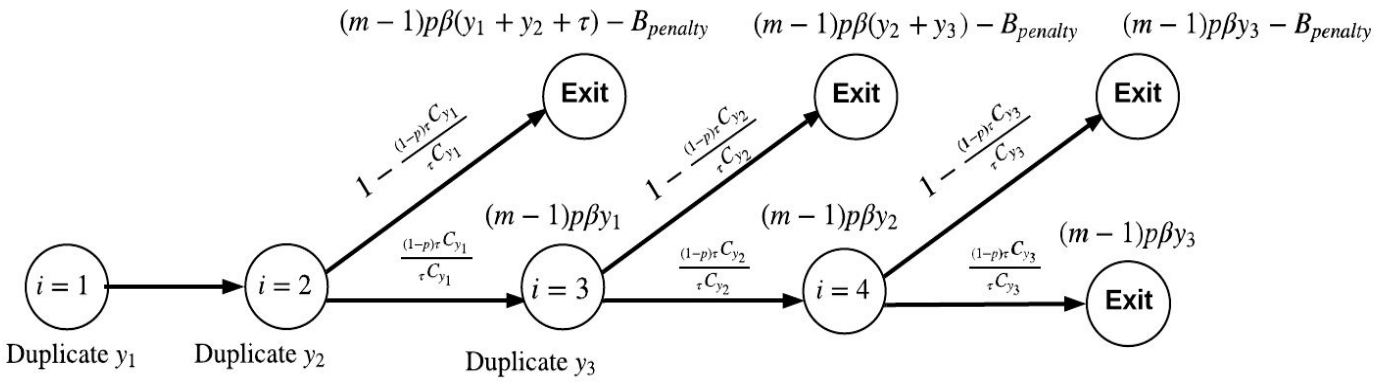
The lottery Protocol

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



Penalty Escrow

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



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:*

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

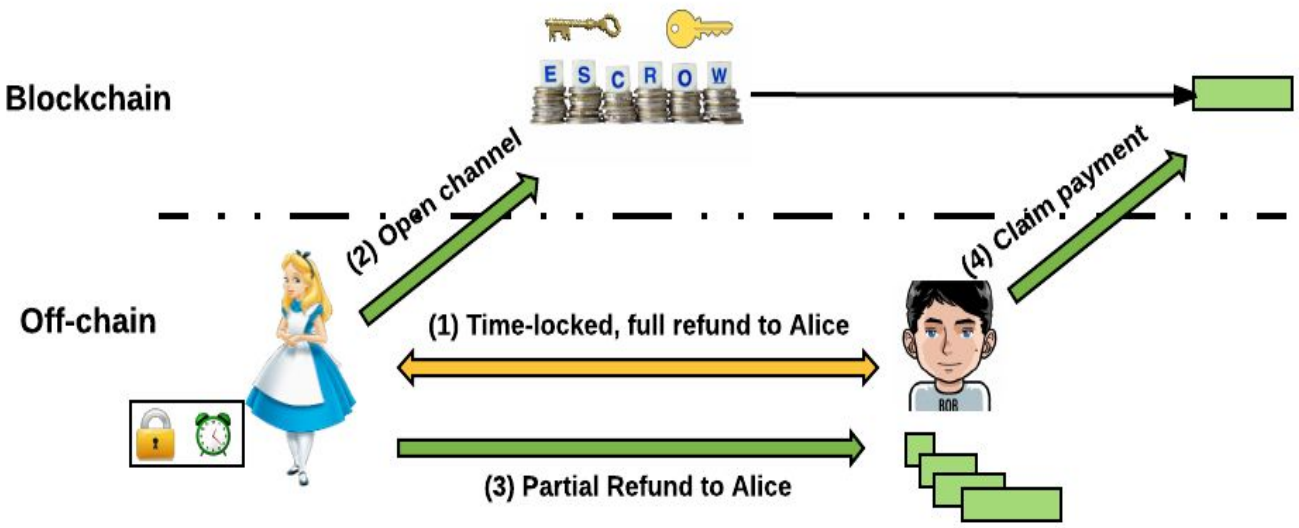
MicroCash - Issues

- 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.

Relation to (Micro)Payment Channels and Networks

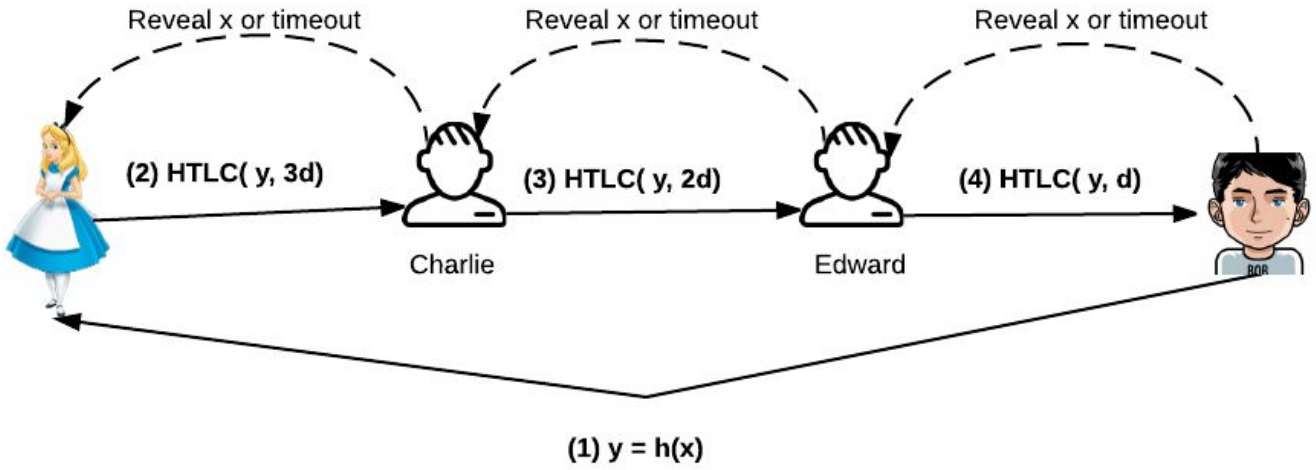
Payment Channels

- A payment chanel is a common locked fund between two parties with the currency ownership adjusted overtime.



Payment Networks

- How about paying several parties using the same escrow?
  - The lightning network [Poon et al., 2014]
  - Alice can pay any 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.

Conclusions

- Micropayments provide a flexible payment paradigm.
  - Reduce risks of payment-service exchange.
  - Allow starting/stopping the service at anytime.
  - Large variety of application, especially rewarding peers in P2P service systems.
- Cryptocurrencies provide useful tools to build fully distributed micropayment schemes.

References

[Wheeler, 1996] David Wheeler. “Transactions using bets.” In International Workshop on Security Protocols, 1996.

[Rivest, 1997] Ronald Rivest. “Electronic lottery tickets as micropayments.” In Financial Cryptography, 1997.

[Pass et al., 2015] Rafael Pass et al. "Micropayments for decentralized currencies." In CCS, 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.