He-HTLC: Revisiting Incentives in HTLC

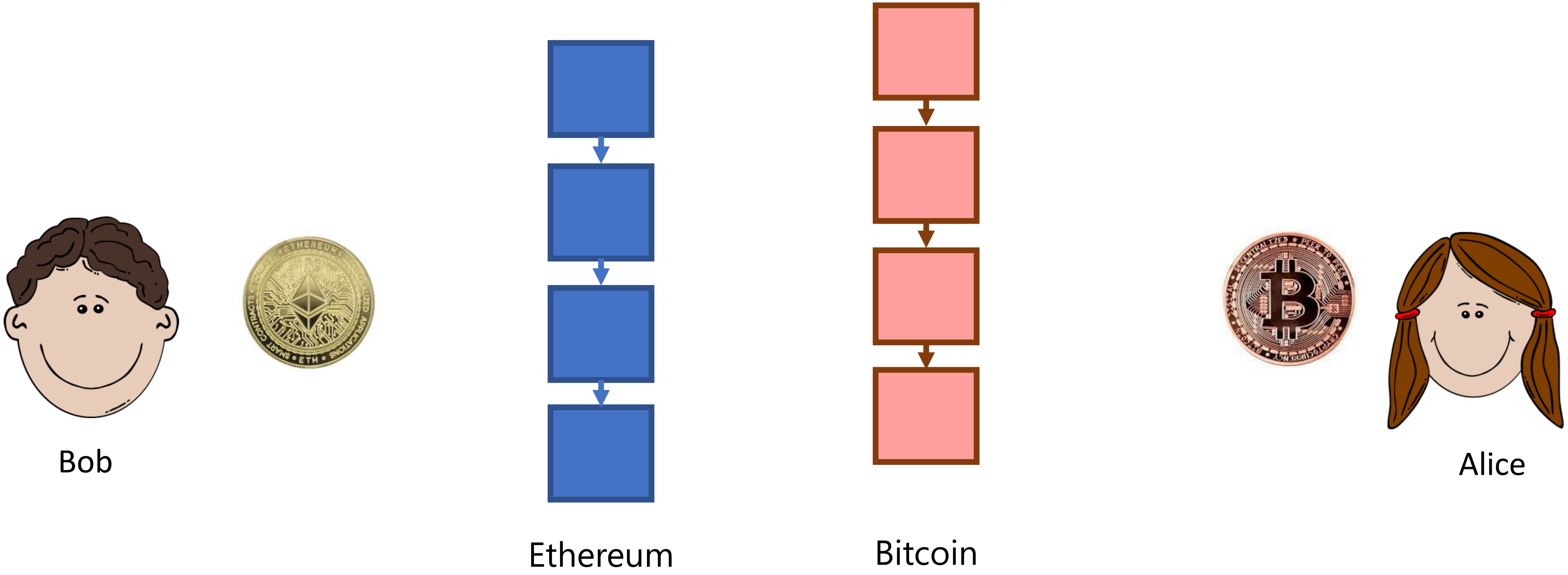
Sarisht Wadhwa

Joint work with Jannis Stöter, Fan Zhang, Kartik Nayak

Duke UNIVERSITY

Cross-Chain Atomic Swap

Aim: Exchange assets on Chain 1 for some assets on Chain 2



Bob

Ethereum

Bitcoin

Alice

HTLC: Hashed Time Lock Contract

Reveal secret to get paid

If no one releases secret until timeout, then refund.

HTLC: Hashed Time Lock Contract

Reveal secret to get paid

If no one releases secret until timeout, then refund.

Deposit/create

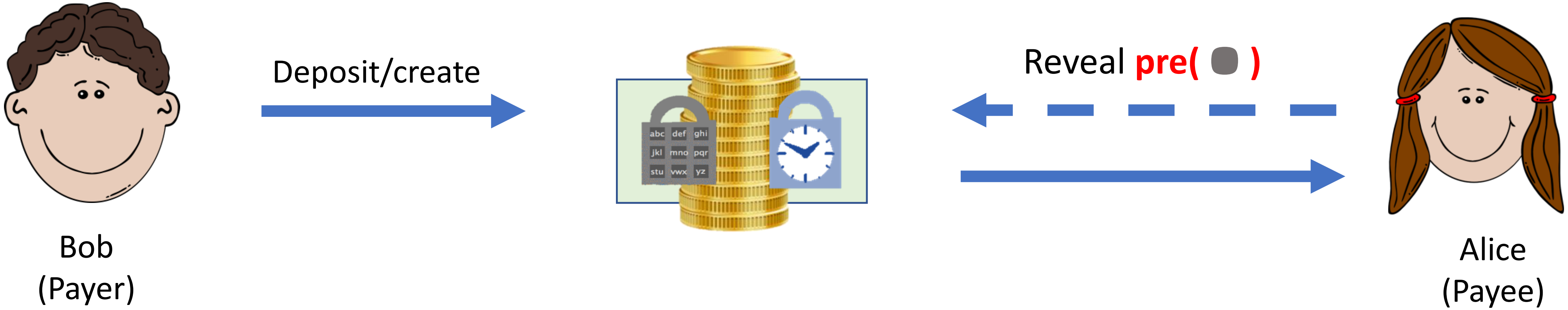
Bob
(Payer)

Alice
(Payee)

HTLC: Hashed Time Lock Contract

Reveal secret to get paid

If no one releases secret until timeout, then refund.



Deposit/create

Reveal pre(●)

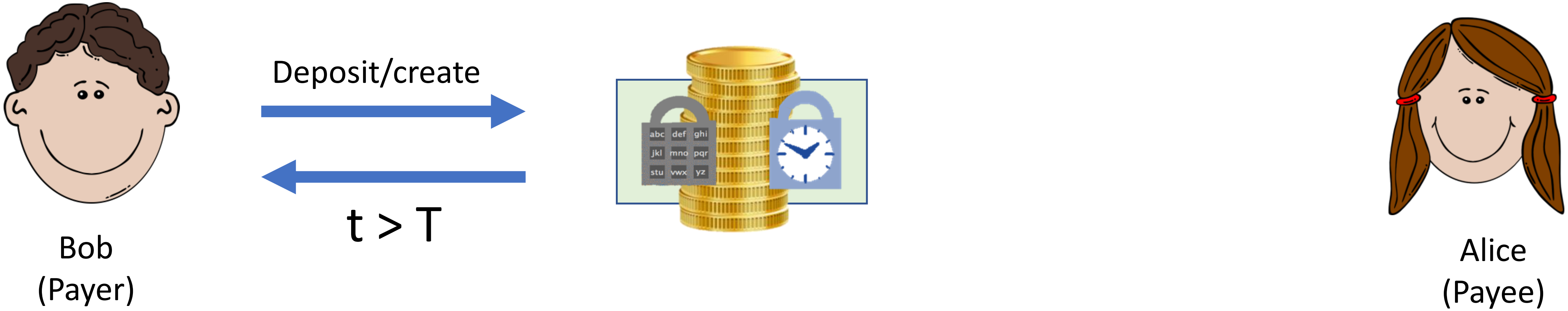
Bob
(Payer)

Alice
(Payee)

HTLC: Hashed Time Lock Contract

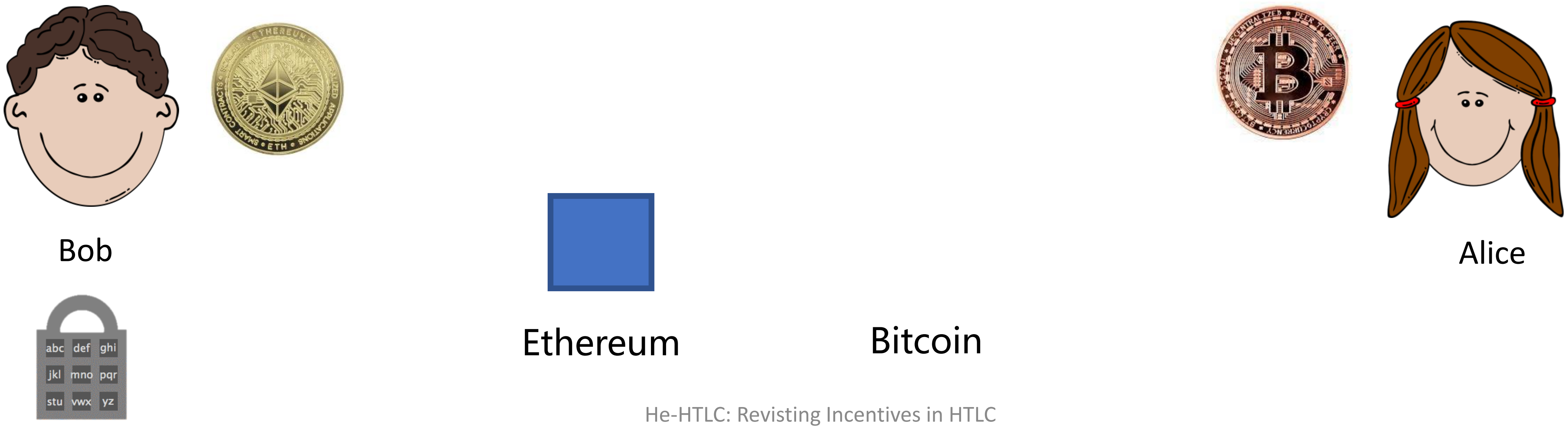
- Reveal secret to get paid

- If no one releases secret until timeout, then refund.



Cross-Chain Atomic Swap

Both lock their assets in HTLCs using a common hashlock



Bob

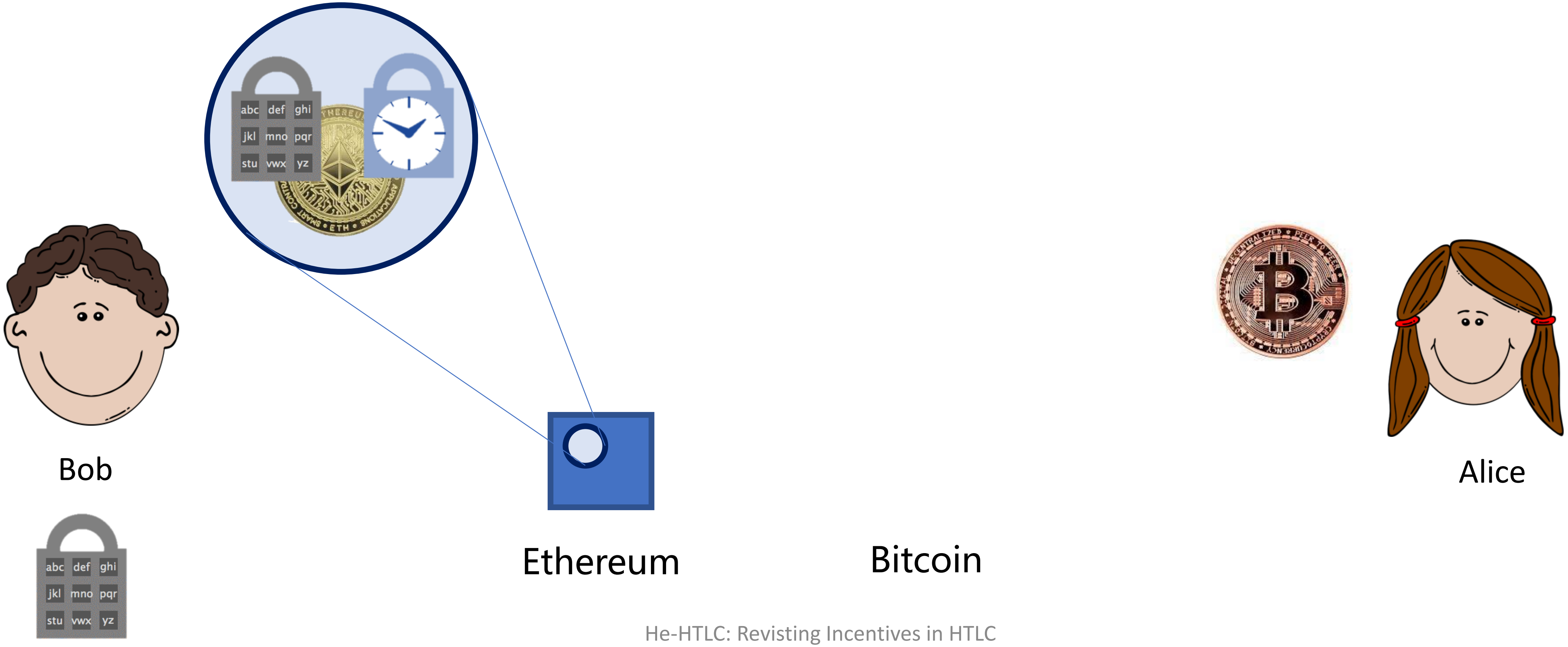
Alice

Ethereum

Bitcoin

Cross-Chain Atomic Swap

Both lock their assets in HTLCs using a common hashlock



Bob

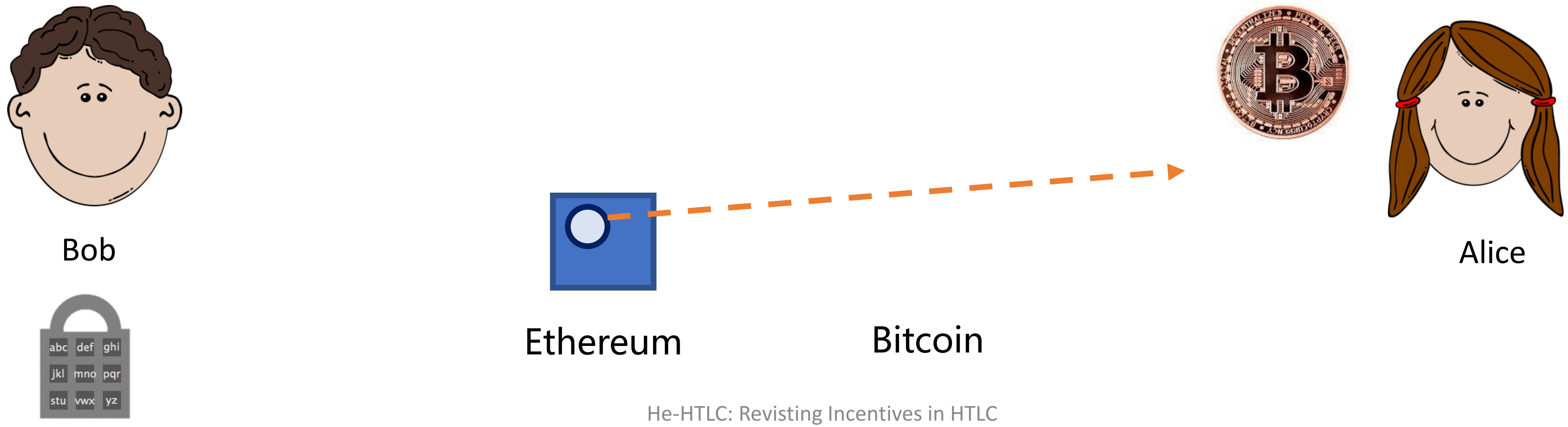
Alice

Ethereum

Bitcoin

Cross-Chain Atomic Swap

Both lock their assets in HTLCs using a common hashlock



Bob

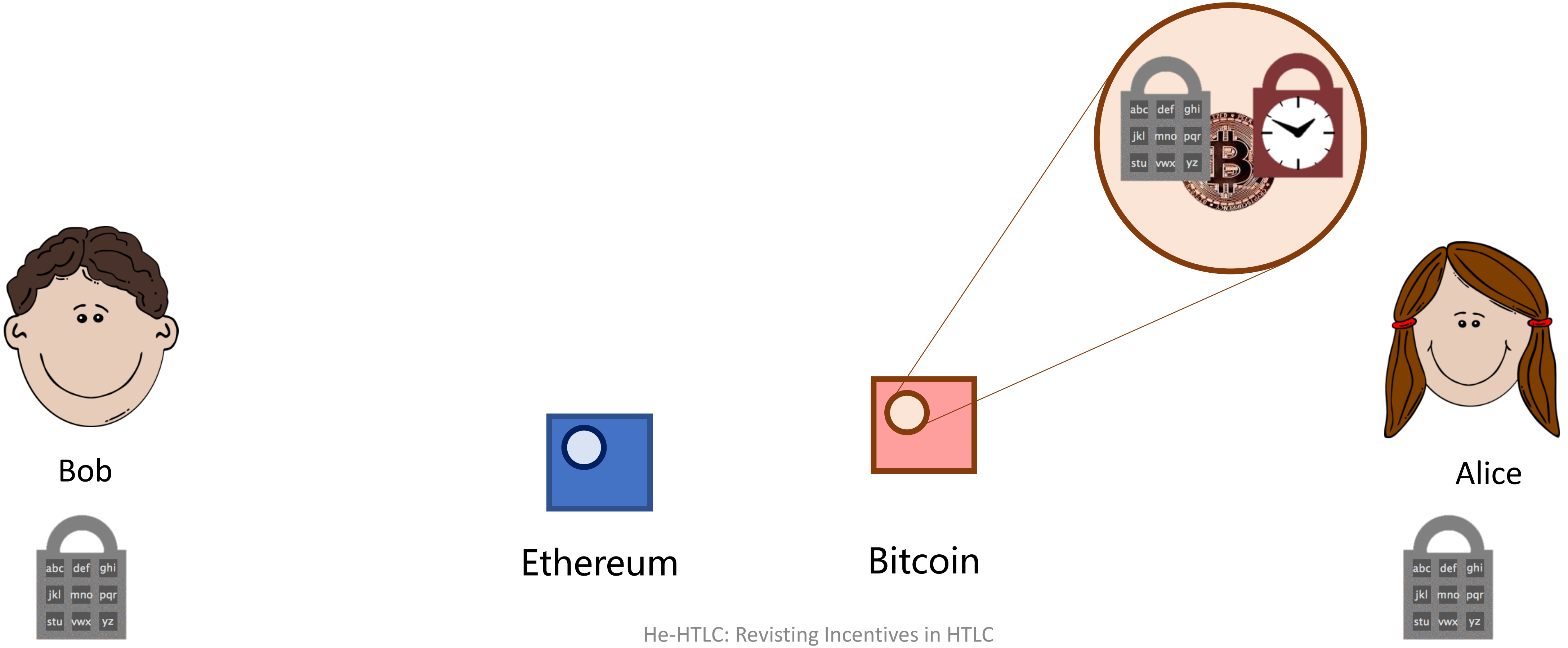
Ethereum

Bitcoin

Alice

Cross-Chain Atomic Swap

Both lock their assets in HTLCs using a common hashlock



Bob

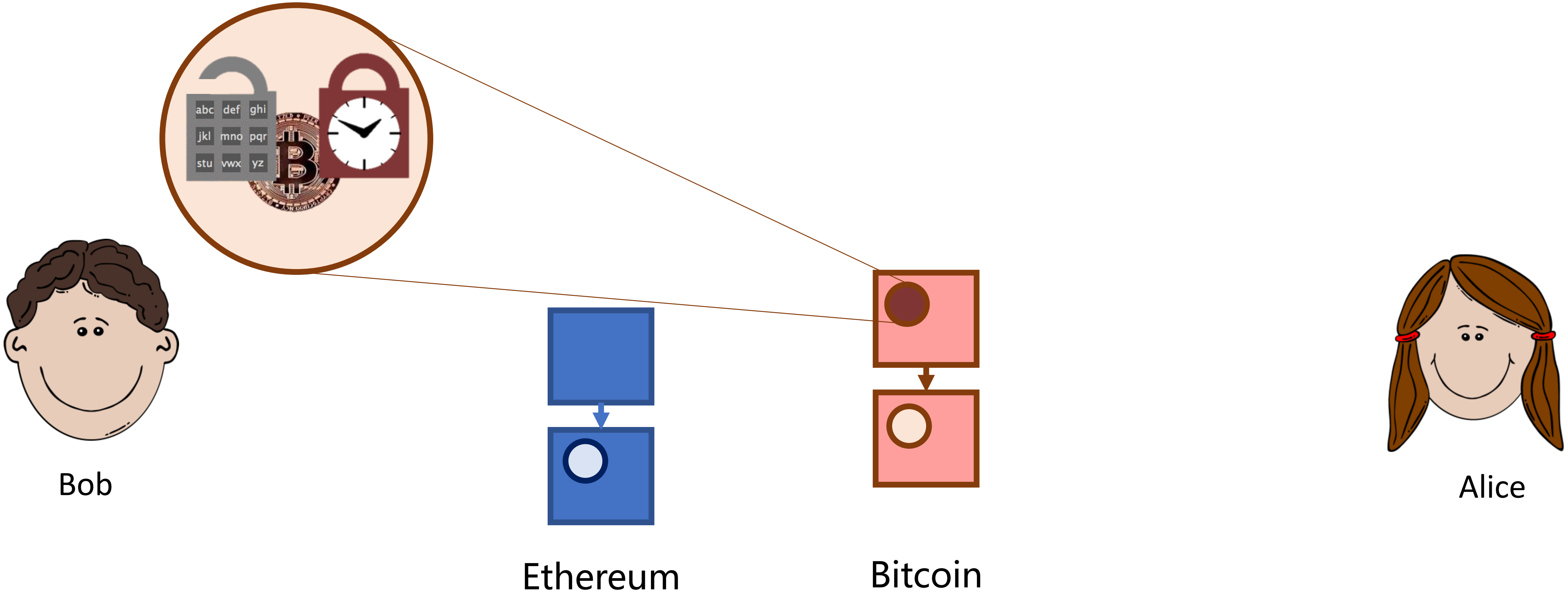
Ethereum

Bitcoin

Alice

Cross-Chain Atomic Swap

Bob knows how to open the hashlock, and does so on Bitcoin



Bob

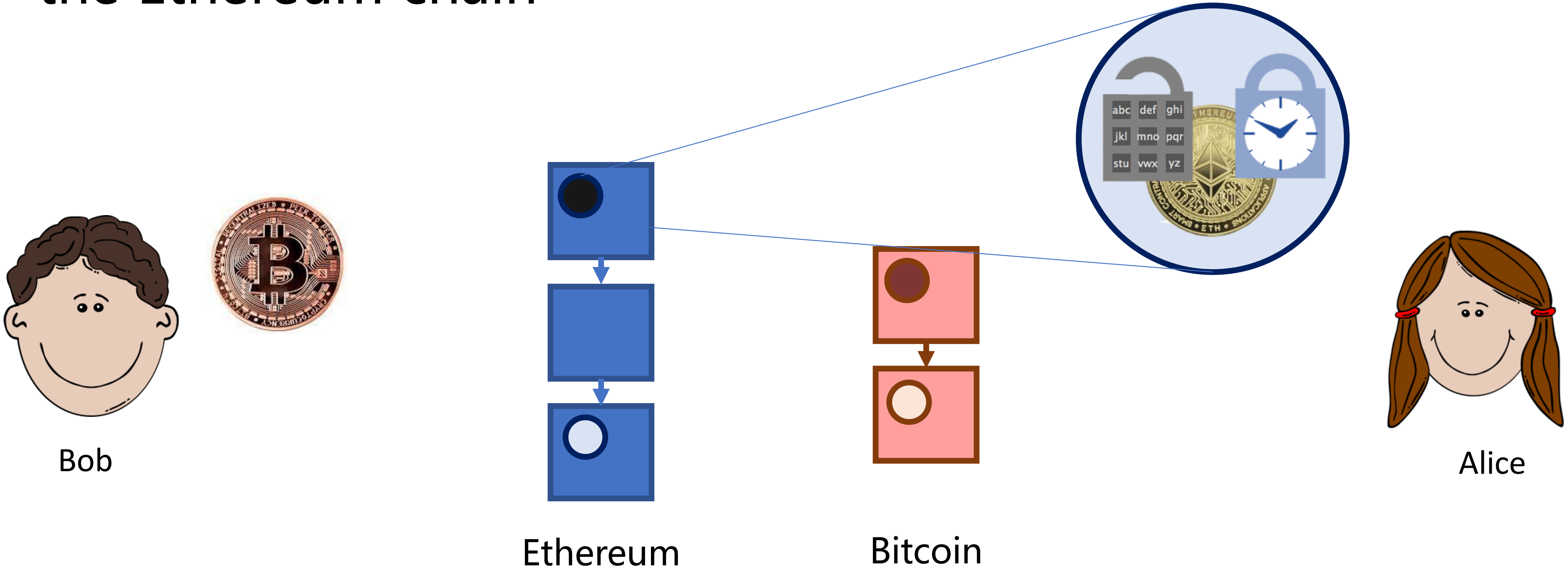
Ethereum

Bitcoin

Alice

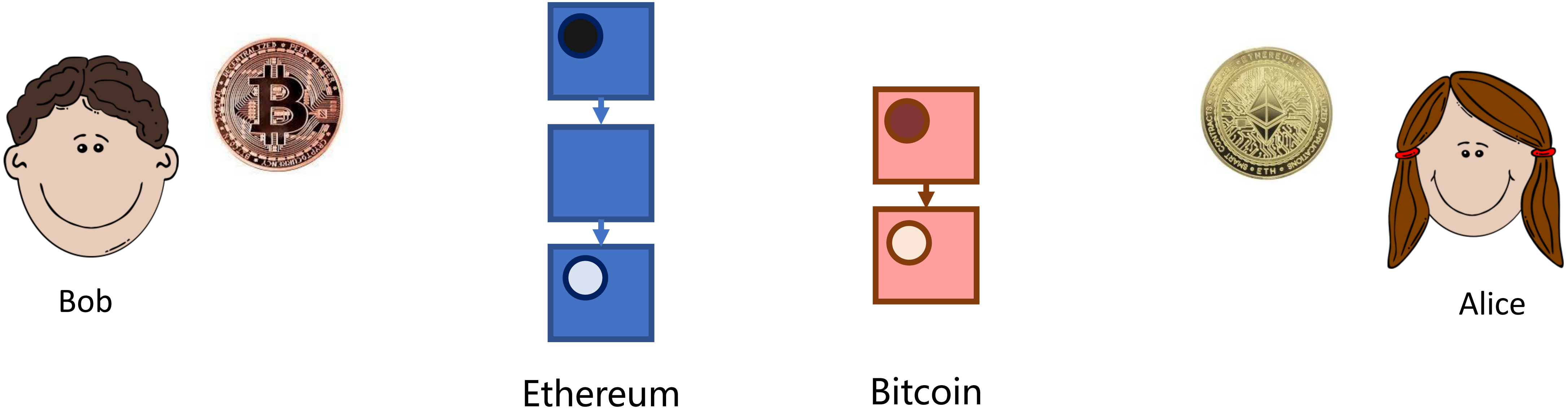
Cross-Chain Atomic Swap

Alice learns how to open the hashlock from Bob, and does so for the Ethereum chain



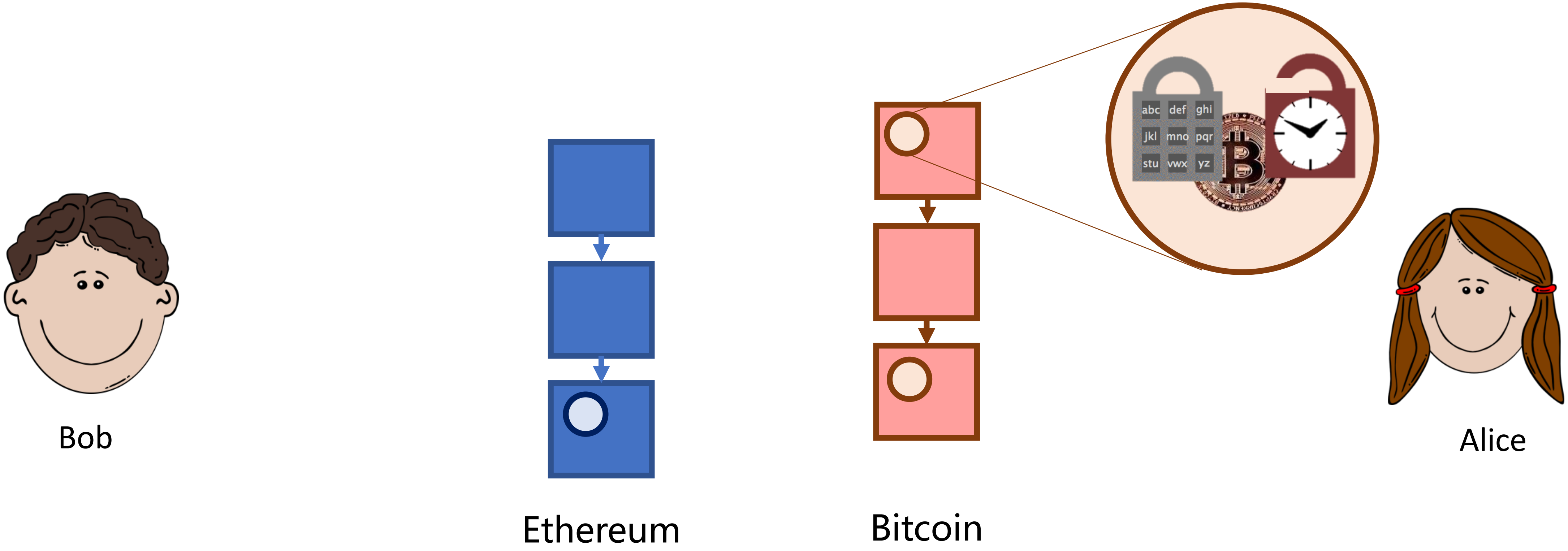
Cross-Chain Atomic Swap

Alice learns how to open the hashlock from Bob, and does so for the Ethereum chain



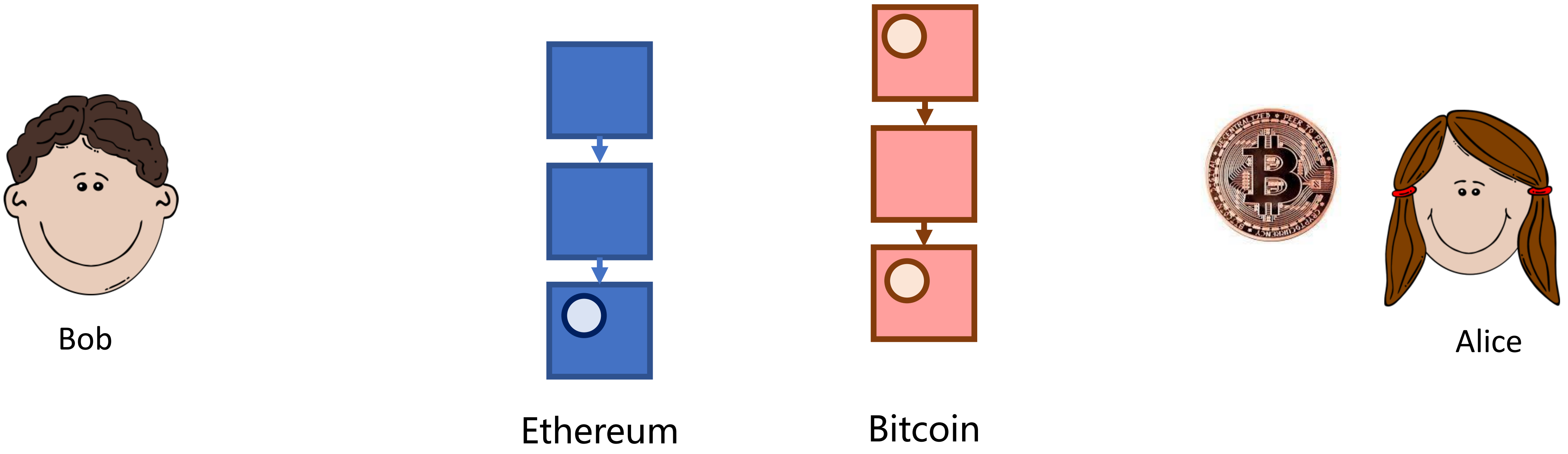
Cross-Chain Atomic Swap

If Bob doesn’t reveal the hashlock, then first, timelock on Alice’s contract expires.



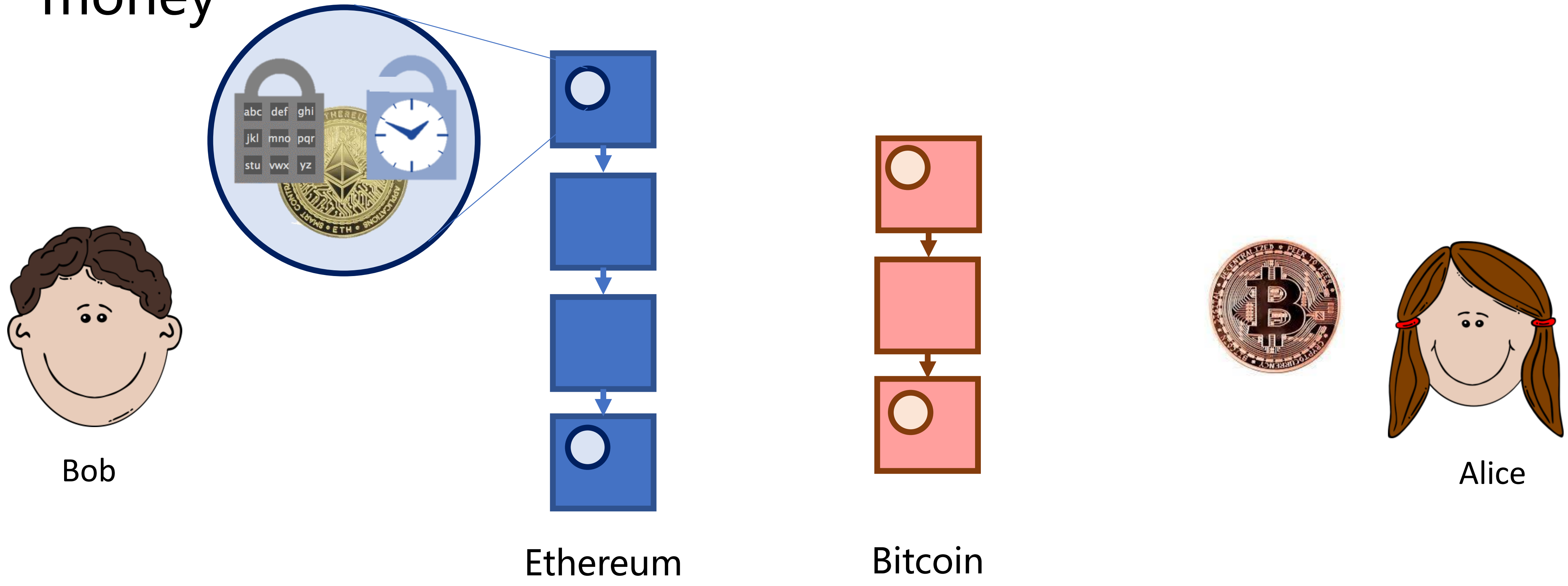
Cross-Chain Atomic Swap

If Bob doesn’t reveal the hashlock, then first, timelock on Alice’s contract expires.



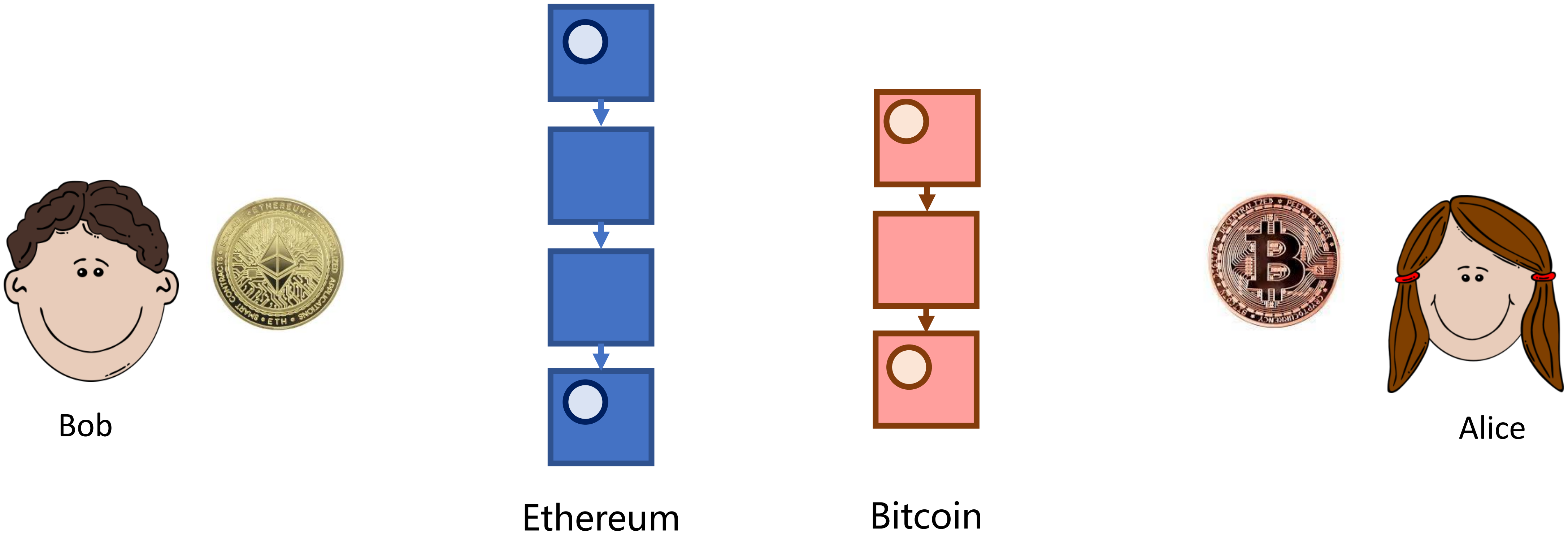
Cross-Chain Atomic Swap

Eventually, the other timelock also expires, and Bob gets back the money



Cross-Chain Atomic Swap

Eventually, the other timelock also expires, and Bob gets back the money



Bob

Ethereum

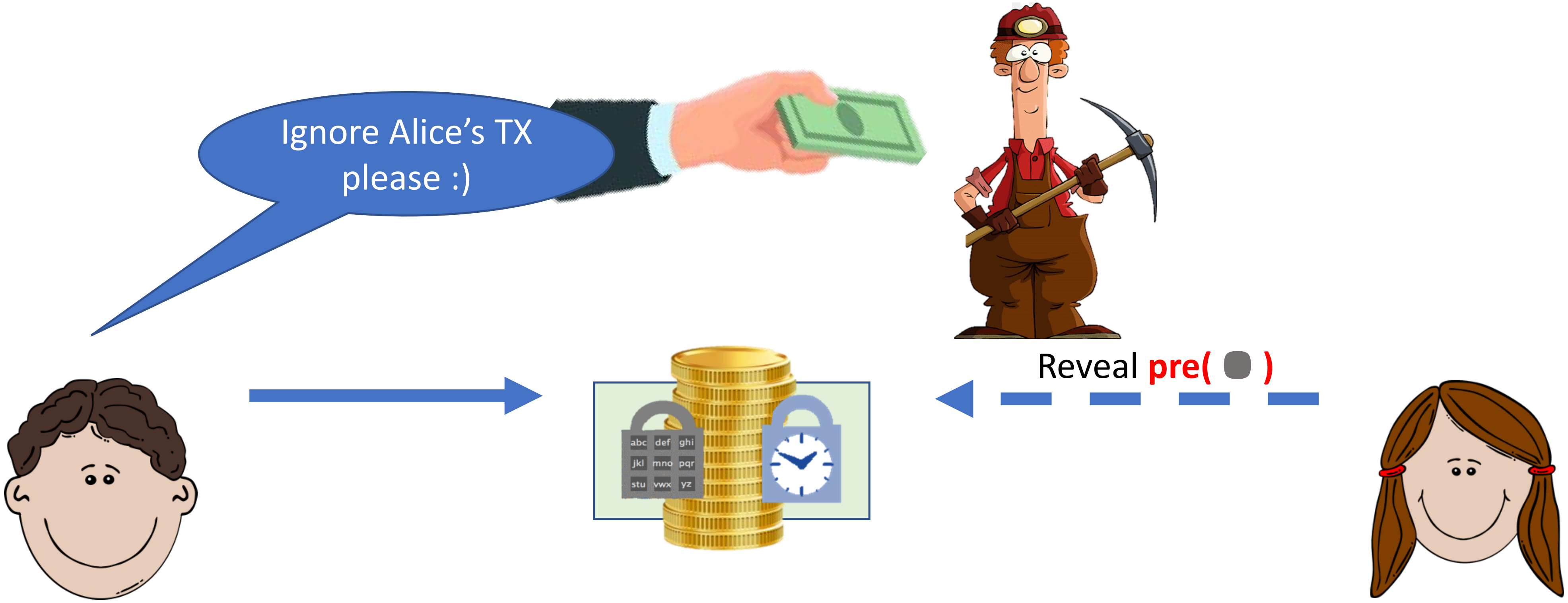
Bitcoin

Alice

Bribery: A Problem with HTLC [HZ’20, WHF’19]

Reveal pre(●)
Bob
Alice

Bribery: A Problem with HTLC [HZ'20, WHF'19]



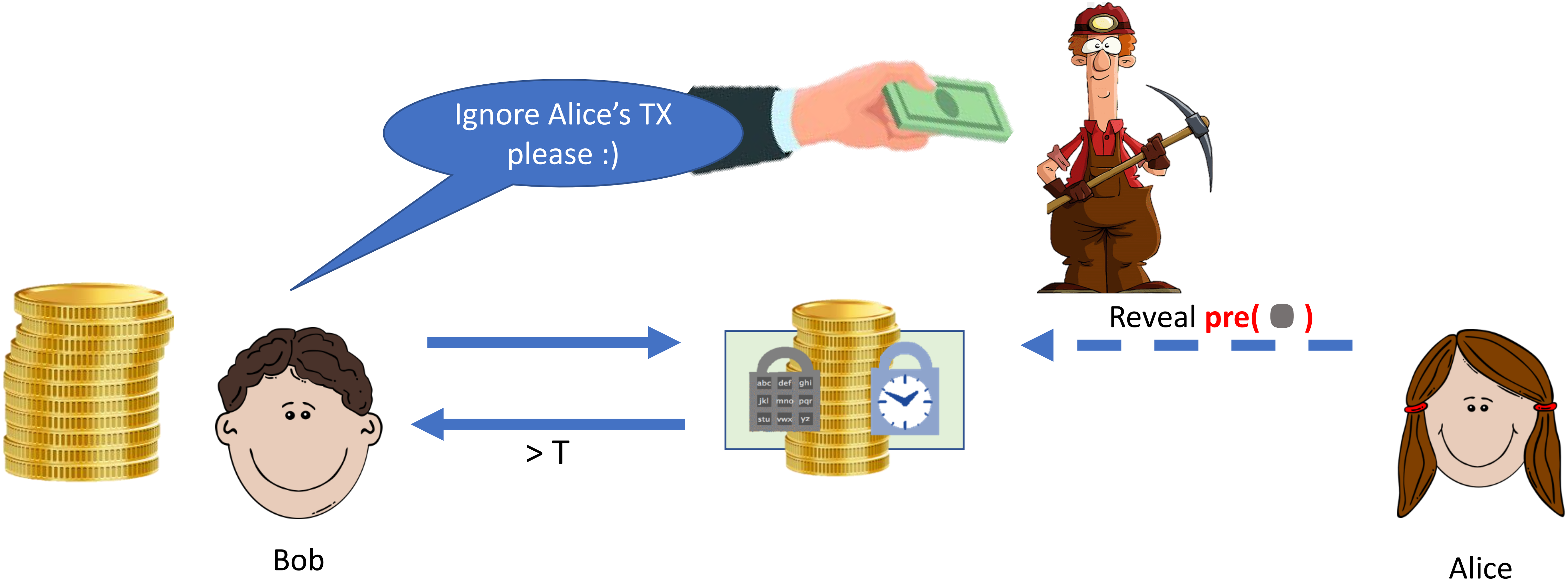
Ignore Alice's TX please :)

Reveal pre(●)

Bob

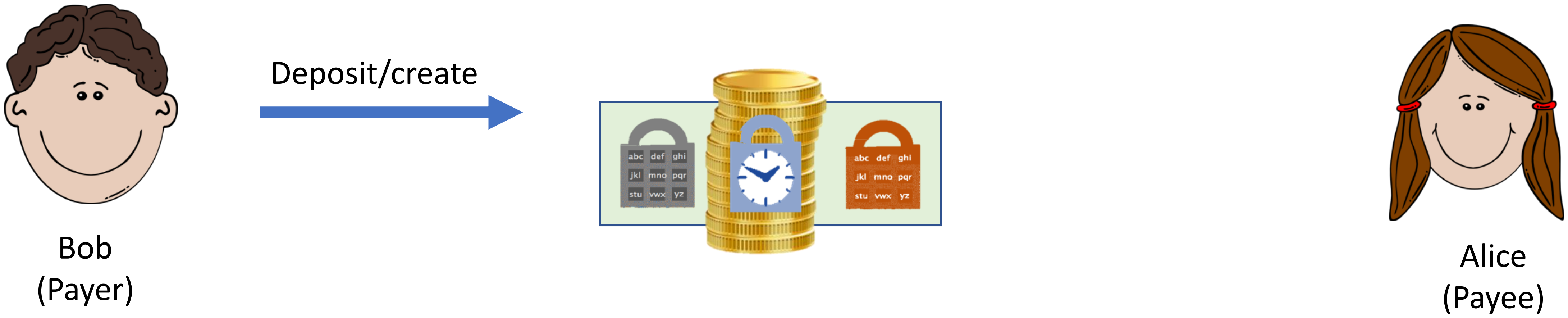
Alice

Bribery: A Problem with HTLC [HZ’20, WHF’19]



MAD-HTLC [TYME'21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



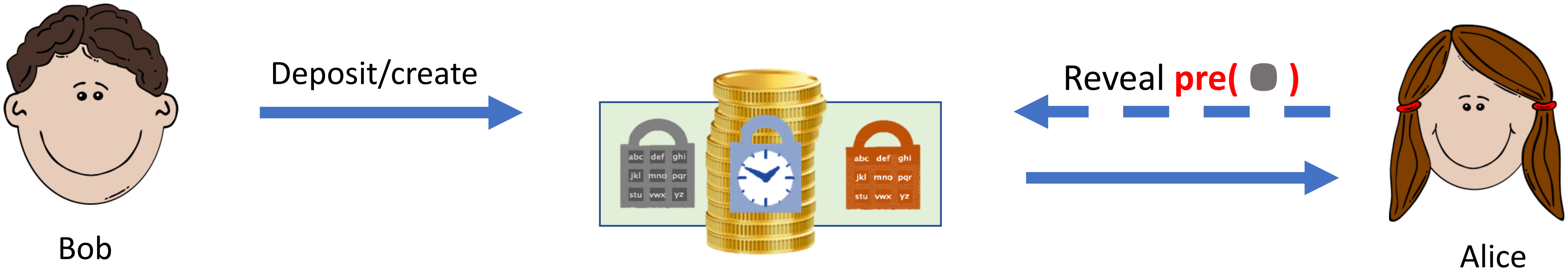
Deposit/create

Bob
(Payer)

Alice
(Payee)

MAD-HTLC [TYME’21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



Deposit/create

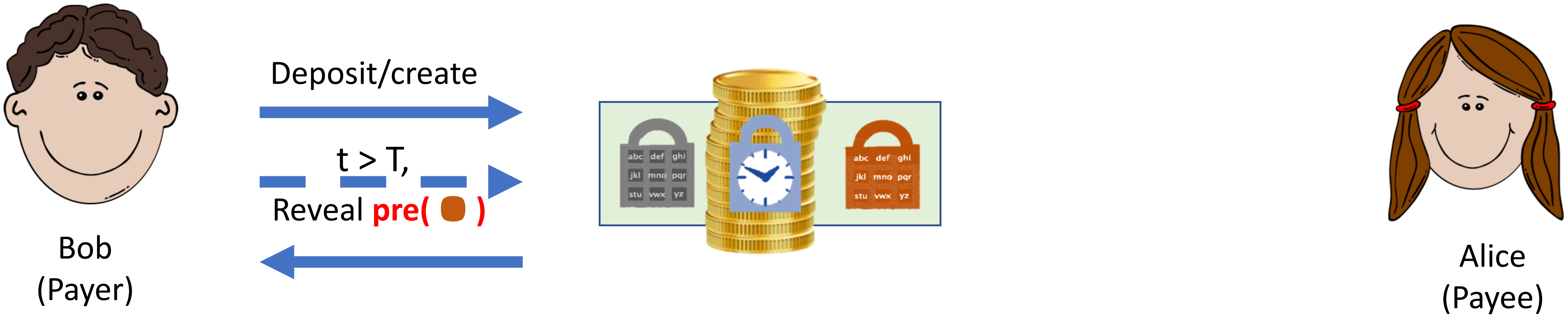
Reveal **pre(■)**

Bob
(Payer)

Alice
(Payee)

MAD-HTLC [TYME'21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



Deposit/create

t > T,

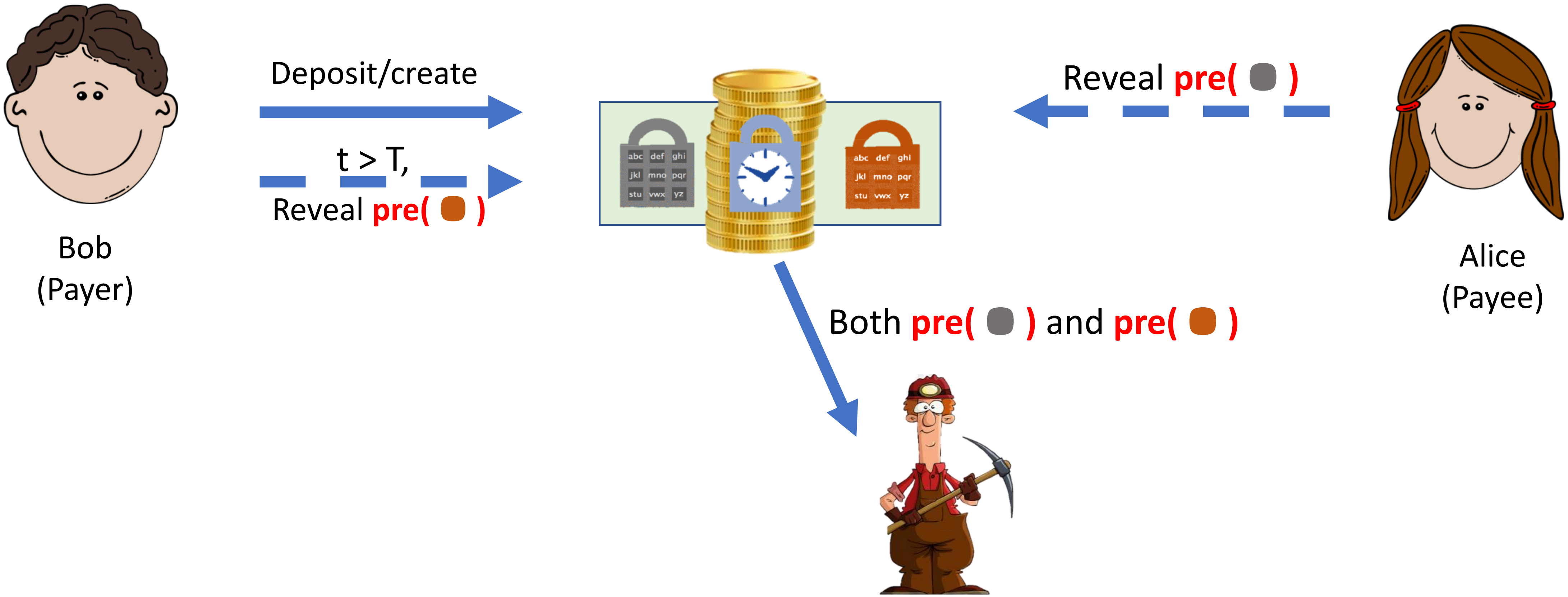
Reveal **pre(●)**

Bob
(Payer)

Alice
(Payee)

MAD-HTLC [TYME'21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



Deposit/create

t > T,

Reveal **pre(🟧)**

Bob
(Payer)

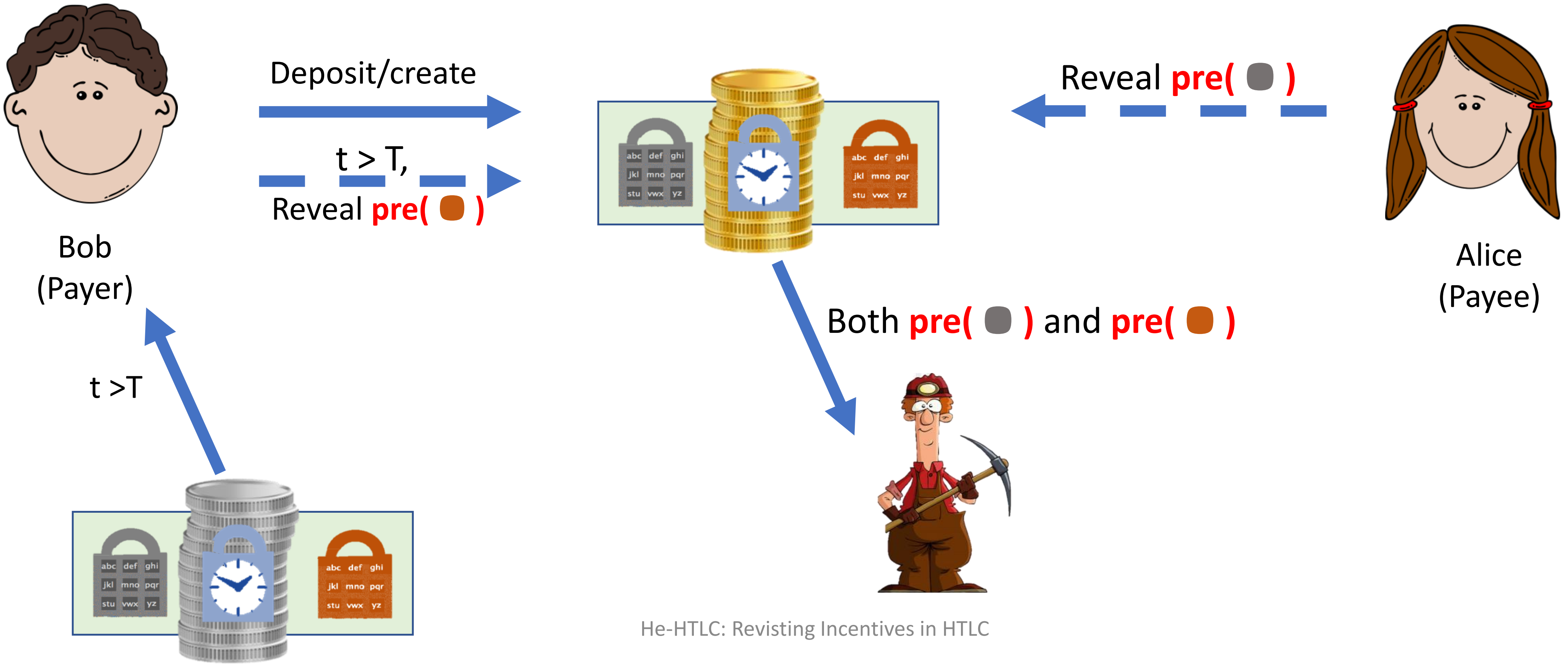
Reveal **pre(⬛)**

Alice
(Payee)

Both **pre(⬛)** and **pre(🟧)**

MAD-HTLC [TYME’21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



Deposit/create

t > T,

Reveal **pre(🟠)**

Reveal **pre(●)**

Bob

(Payer)

Alice

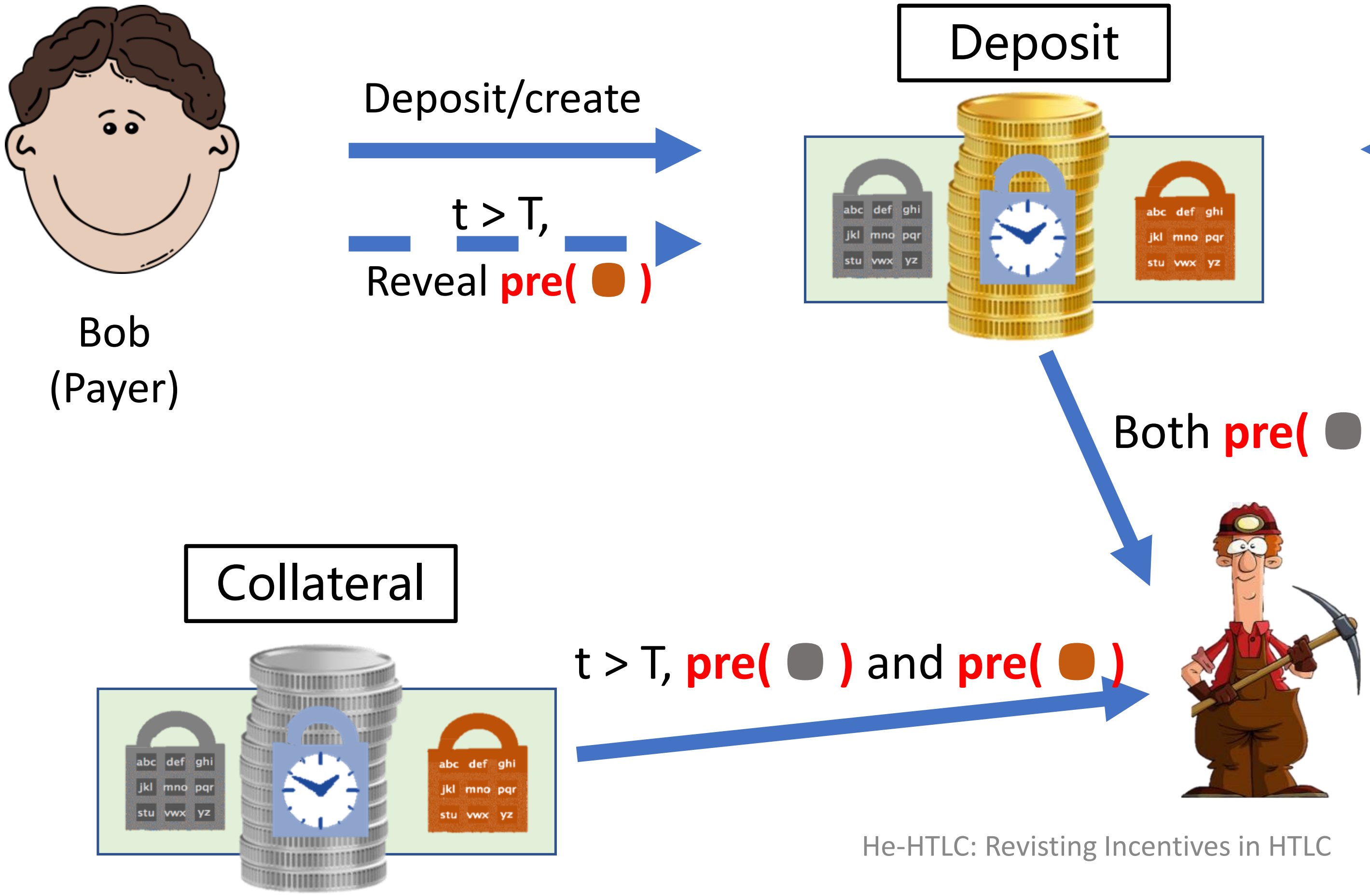
(Payee)

Both **pre(●)** and **pre(🟠)**

t >T

MAD-HTLC [TYME’21]

Make both Alice and Bob lose if anyone cheats – *Mutually Assured Destruction*



Bob
(Payer)

Deposit/create

t > T,
Reveal **pre(🟠)**

Deposit

Reveal **pre(●)**

Alice
(Payee)

Both **pre(●)** and **pre(🟠)**

Collateral

t > T, **pre(●)** and **pre(🟠)**

This **defeats** Bob’s bribery because:

- Bob must reveal **pre(🟠)** to realize profit and then miners will grab everything.

Contributions: Revisiting Incentives in HTLC

Attacks on HTLC Schemes

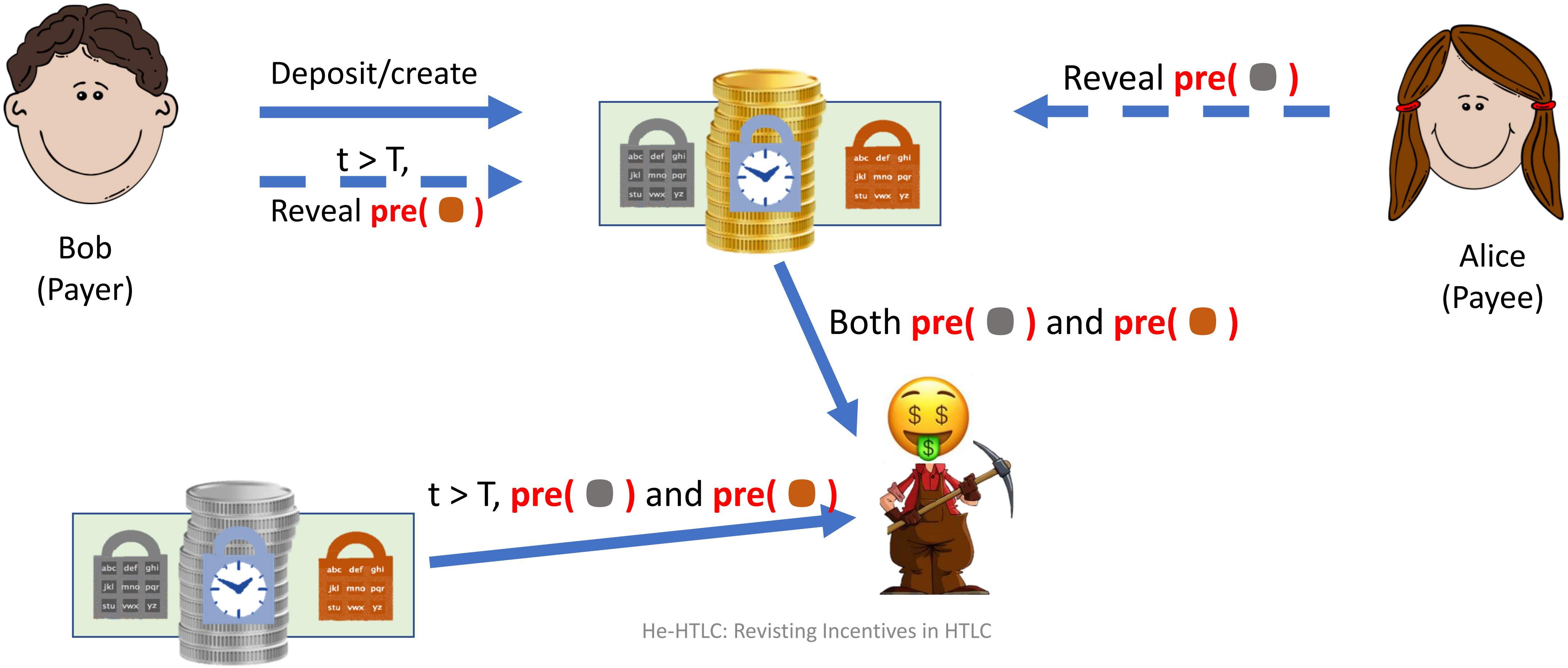
- Notion of actively rational miners
- Three reverse bribery attacks (RBA)
  - Success Independent RBA
  - Success Dependent RBA
  - Hybrid Attack

He-HTLC

An incentive-compatible HTLC scheme

MAD-HTLC: Is it Safe?

For a miner, achieving the following state is the best-case scenario.



Bob
(Payer)

Deposit/create

t > T,
Reveal **pre(🟧)**

Reveal **pre(⬛)**

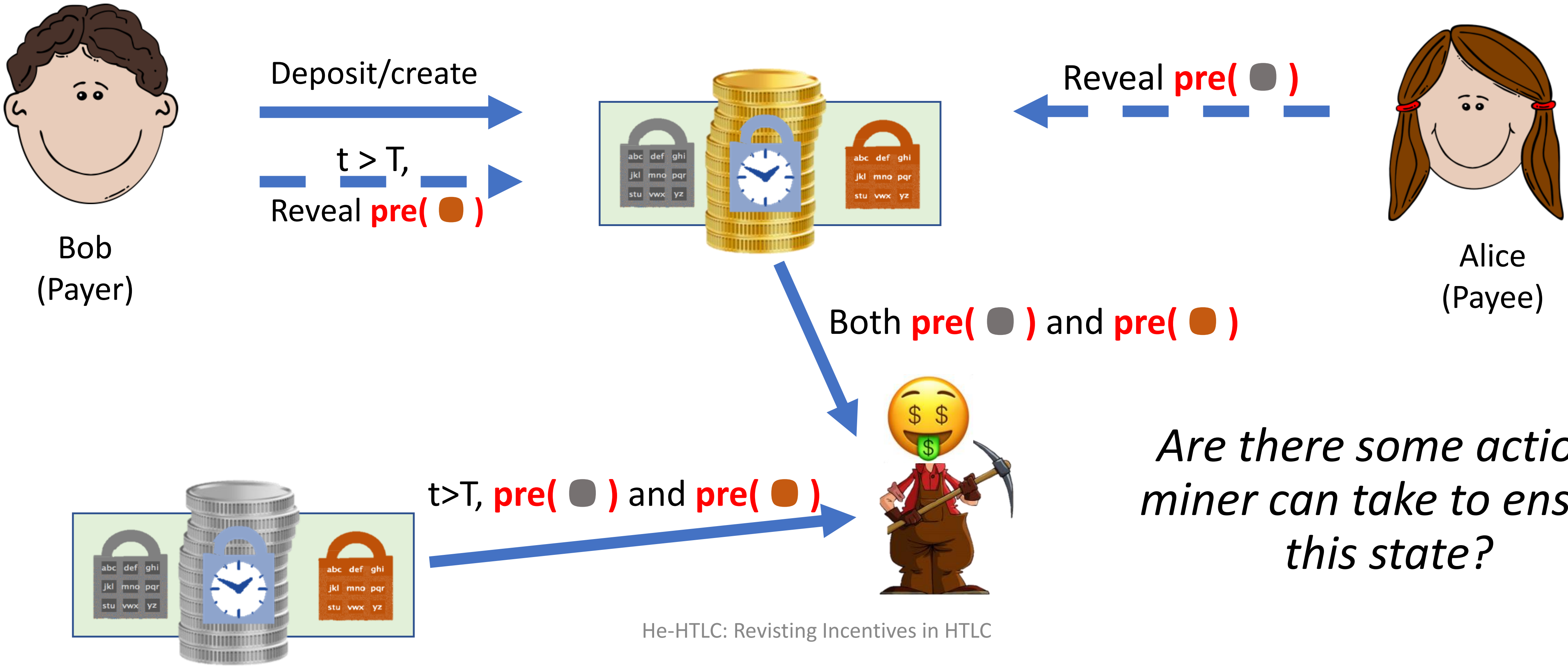
Alice
(Payee)

Both **pre(⬛)** and **pre(🟧)**

t > T, **pre(⬛)** and **pre(🟧)**

MAD-HTLC: Is it Safe?

For a miner, achieving the following state is the best-case scenario.



Bob
(Payer)

Deposit/create

t > T,
Reveal **pre(●)**

Reveal **pre(●)**

Alice
(Payee)

Both **pre(●)** and **pre(●)**

t>T, **pre(●)** and **pre(●)**

Are there some actions miner can take to ensure this state?

Passive vs Active Miners

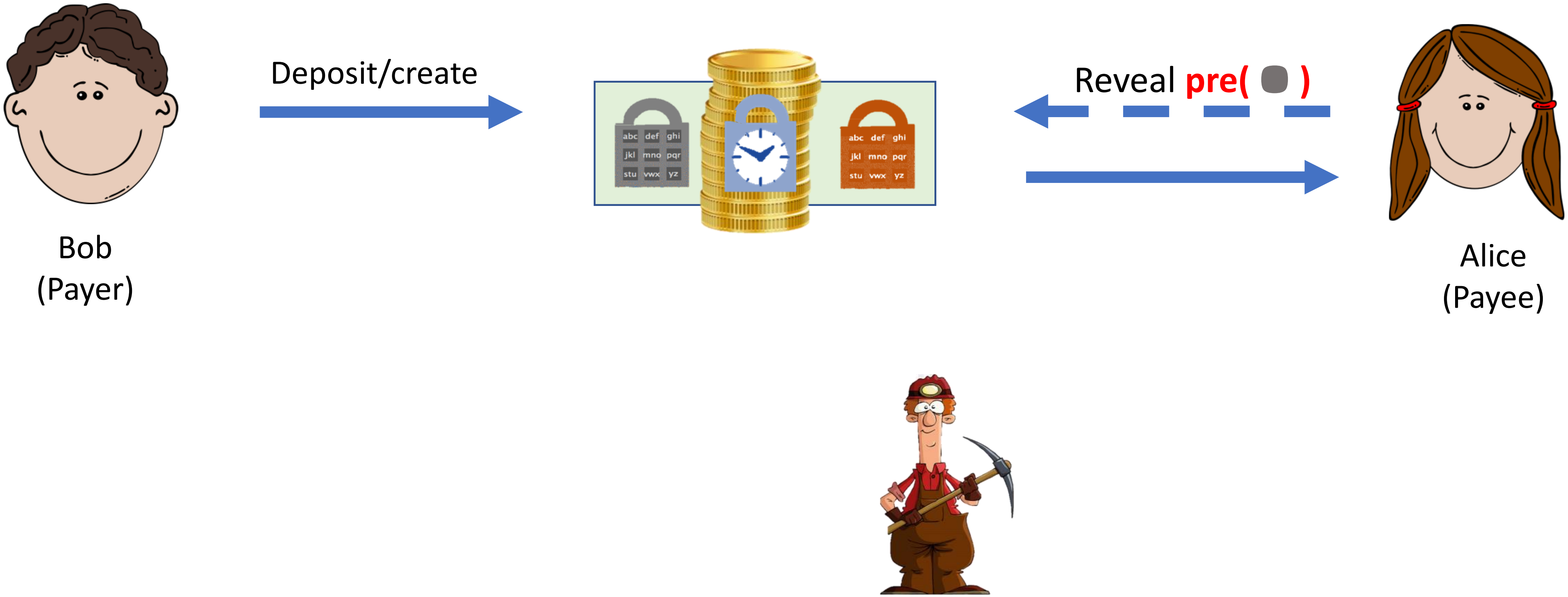
Passive miners

- Focused on the mempool
- Confirming most profitable transactions

Active miners

- Engage in external protocols
- E.g., adding MEV software, open up direct channels to users, etc.

Reverse Bribery: Active Miners’ Action



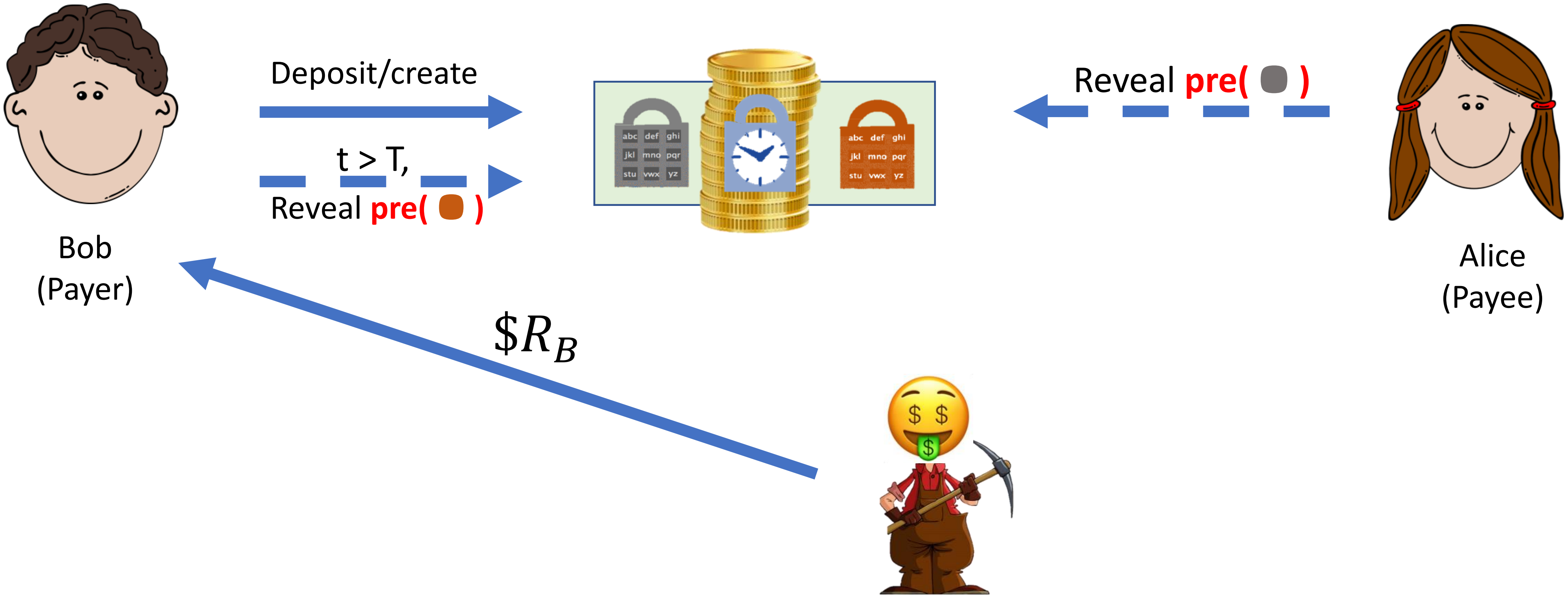
Bob
(Payer)

Deposit/create

Reveal pre(●)

Alice
(Payee)

Reverse Bribery: Active Miners’ Action



Deposit/create

t > T,

Reveal pre(●)

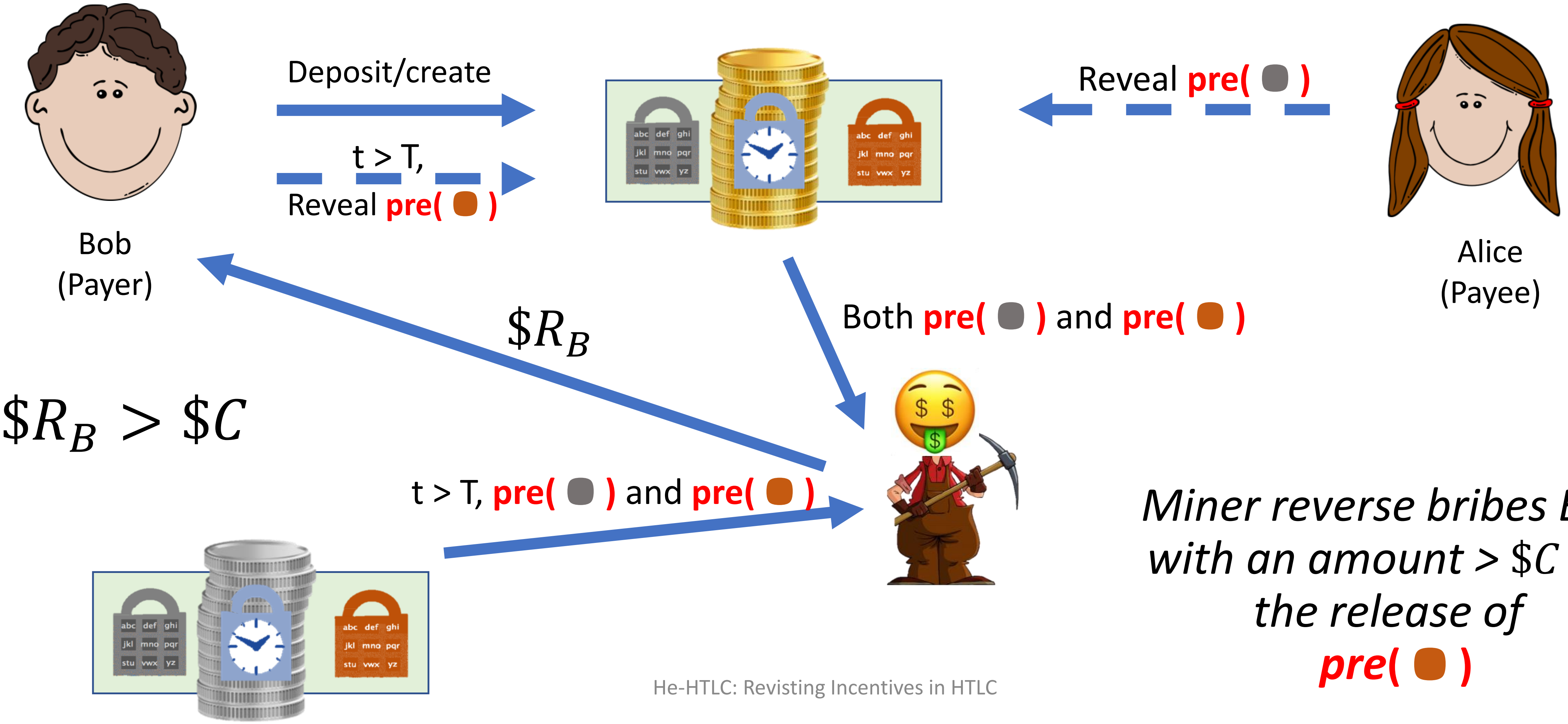
Reveal pre(●)

Bob
(Payer)

Alice
(Payee)

$\$R_B$

Reverse Bribery: Active Miners’ Action

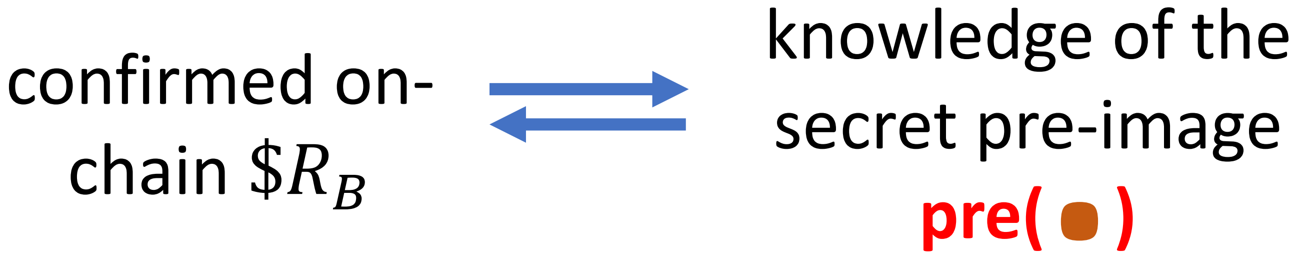


$\$R_B > \$C$

Miner reverse bribes Bob with an amount > $C for the release of ***pre(🟧)***

Attacks Based on Reverse Bribery (RBA)

- Success Independent RBA



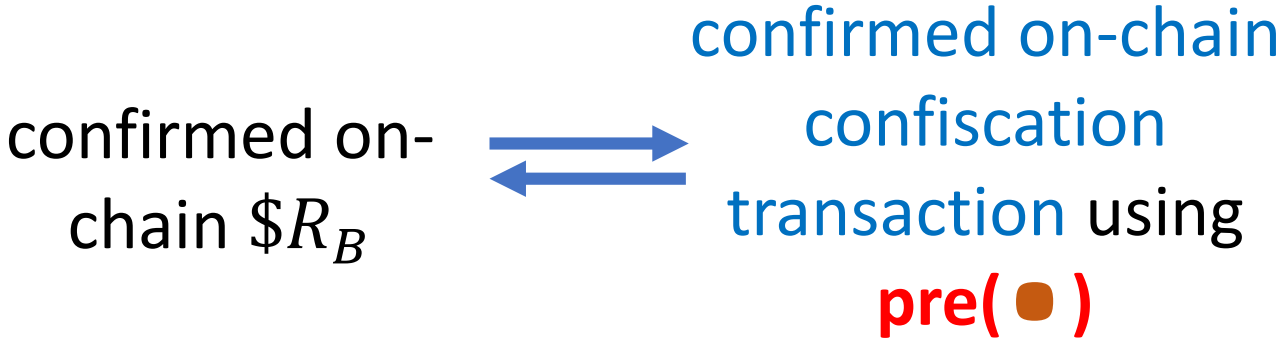
confirmed on-chain $$R_B$

knowledge of the secret pre-image

pre(●)

Attacks Based on Reverse Bribery (RBA)

- Success Independent RBA
- Success Dependent RBA

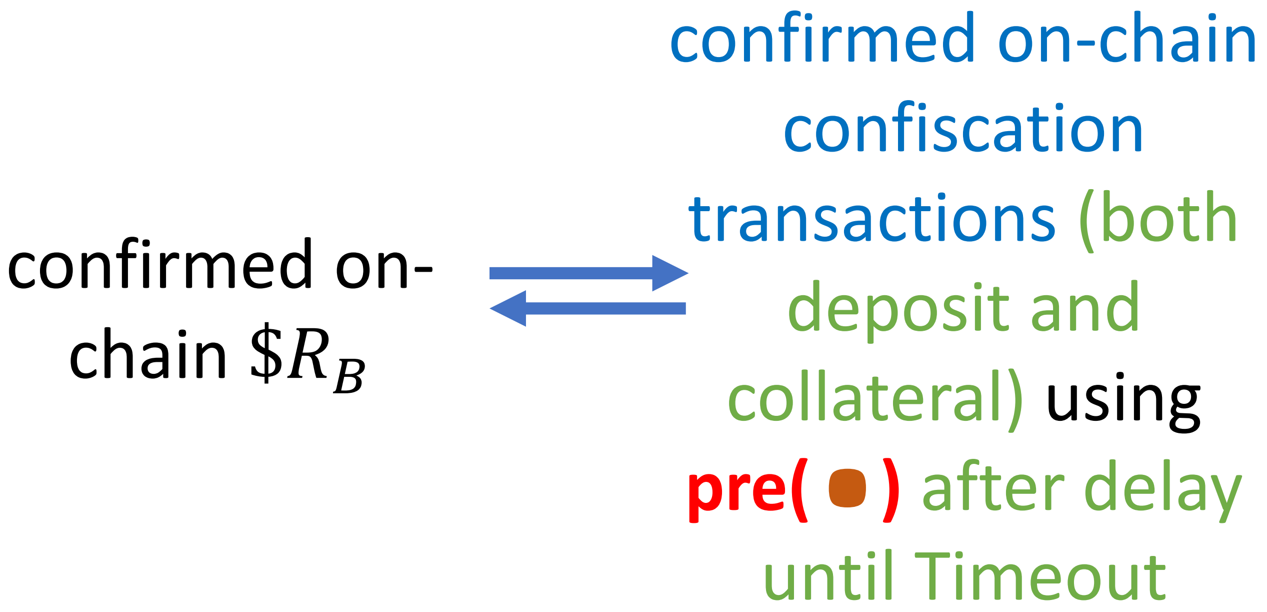


confirmed on-chain $\$R_B$

confirmed on-chain confiscation transaction using **pre(●)**

Attacks Based on Reverse Bribery (RBA)

- ❖ Success Independent RBA
- ❖ Success Dependent RBA
- ❖ Hybrid Delay-RBA



confirmed on-chain $\$R_B$

confirmed on-chain confiscation transactions (both deposit and collateral) using **pre(●)** after delay until Timeout

Designing HTLC: Challenges

- **Bribery Resistance:** The payer must have a way to get back all the money ($\$V + \$C$) after the timeout.

Designing HTLC: Challenges

- **Bribery Resistance:** The payer must have a way to get back all the money ($\$V + \$C$) after the timeout.

  😟 Payer must not be able to bribe a miner more than what the miner receives as enforcer.

Designing HTLC: Challenges

- **Bribery Resistance:** The payer must have a way to get back all the money ($\$V + \$C$) after the timeout.

  😟 Payer must not be able to bribe a miner more than what the miner receives as enforcer.

- **Reverse Bribery Resistance:** In MAD-HTLC miner earns too much when punishing bribery attempts.

Designing HTLC: Challenges

➢ **Bribery Resistance:** The payer must have a way to get back all the money ($\$V + \$C$) after the timeout.

😟 Payer must not be able to bribe a miner more than what the miner receives as enforcer.

➢ **Reverse Bribery Resistance:** In MAD-HTLC miner earns too much when punishing bribery attempts.

😟 A miner must receive $\leq \$C$.

Designing HTLC: Key Ideas

- **Bribery Resistance:** The payer must have a way to get back all the money ($\$V + \$C$) after the timeout.
  - 😟 Payer must not be able to bribe a miner more than what the miner receives as enforcer.

- **Reverse Bribery Resistance:** In MAD-HTLC miner earns too much when

💡😃 Burn the deposit ($\$V$) to avoid reverse bribery

Designing HTLC: Key Ideas

- **Bribery Resistance:** The payer must have a way to get back all the

Make payer bribe multiple miners, so that not all of them can be bribed!

- **Reverse Bribery Resistance:** In MAD-HTLC miner earns too much when

Burn the deposit ($\$V$) to avoid reverse bribery

He-HTLC: Anti-Bribery

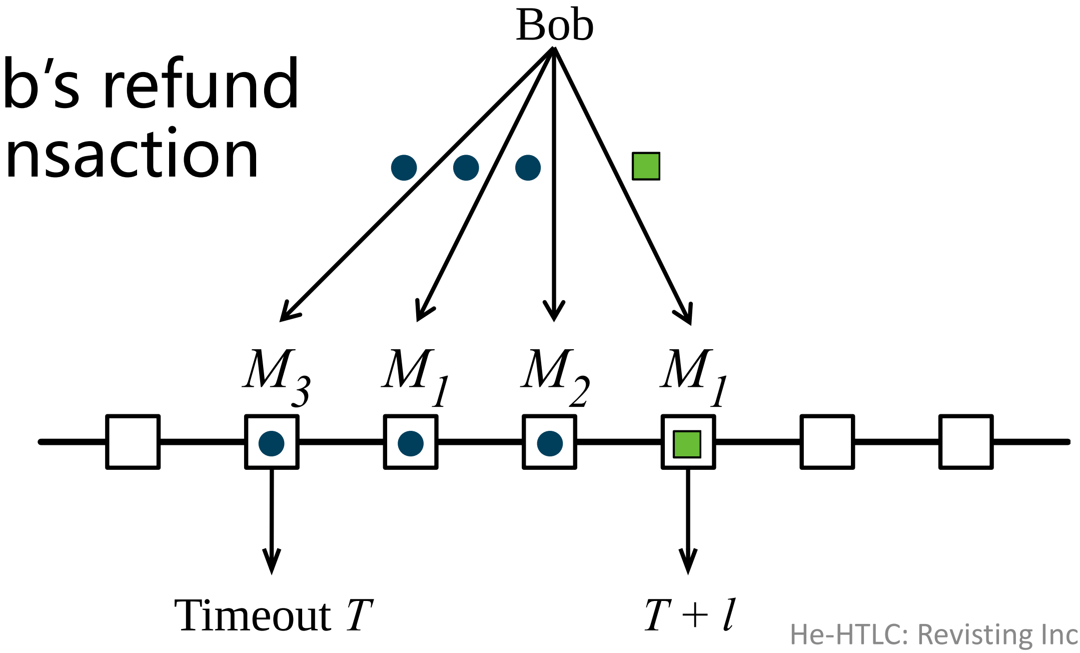
- *Cannot give miner more than* $\$C$ *(Anti-RBA)*

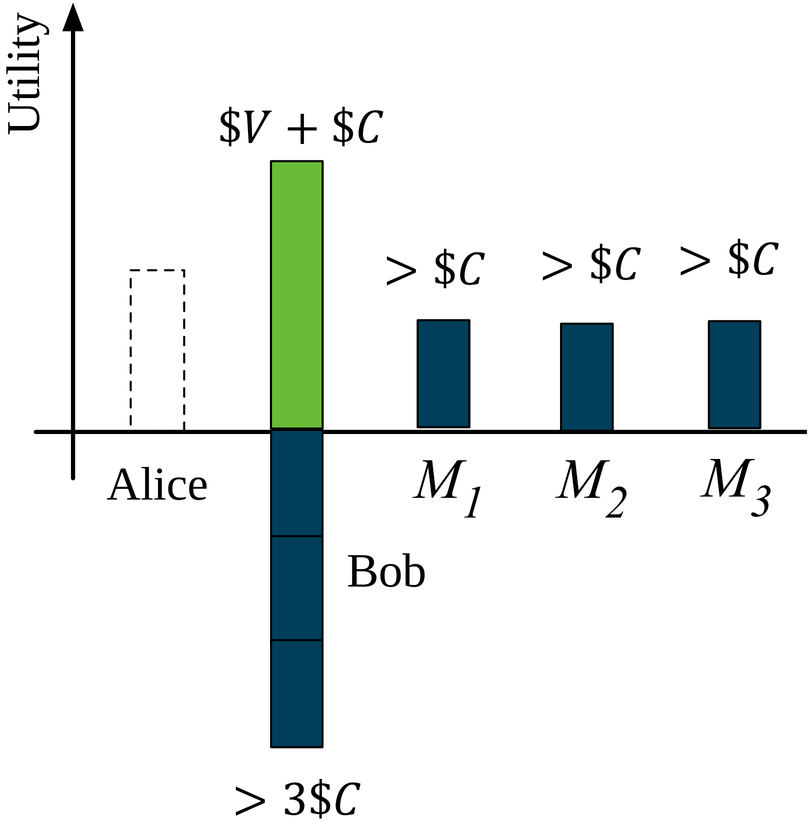
	Miner	***Bob***
Honest	0	$\$C$
Confiscate	$\$C$	0
Get bribe	$\$B$	$\$V + \$C - \$B$

He-HTLC: Anti-Bribery

- *Cannot give miner more than* $\$C$ *(Anti-RBA)*
- *Make Bob to bribe say* $l = 3$ *miners (Anti-Bribery)*

🟩 Bob's refund transaction





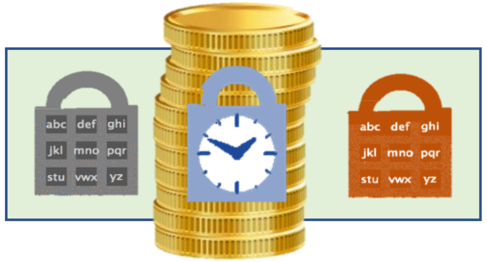
He-HTLC: An Incentive Compatible HTLC

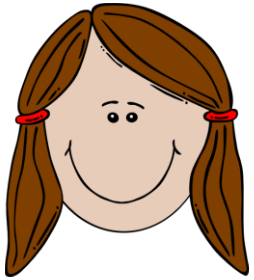
Bob
(Payer)

Deposit/create

($$V + C)

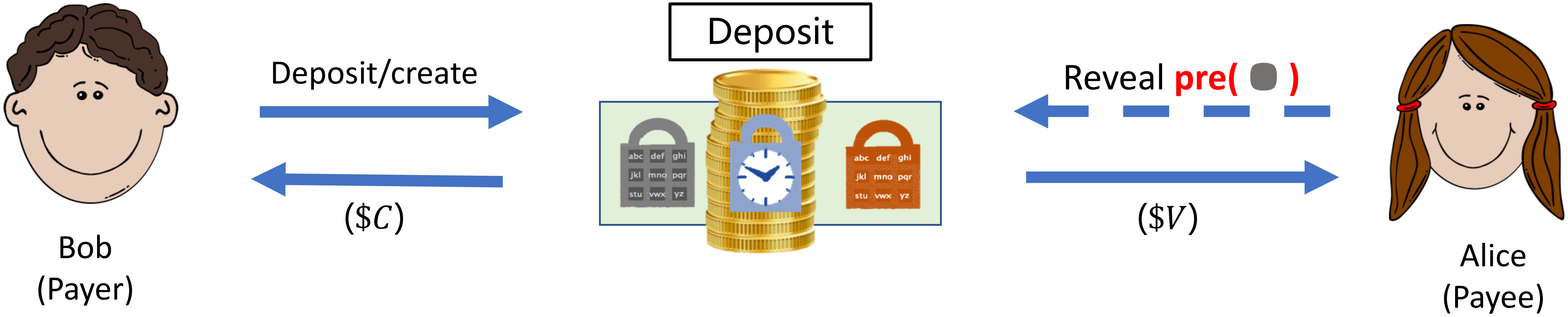
Deposit





Alice
(Payee)

He-HTLC: An Incentive Compatible HTLC



Bob
(Payer)

Deposit/create

(C)

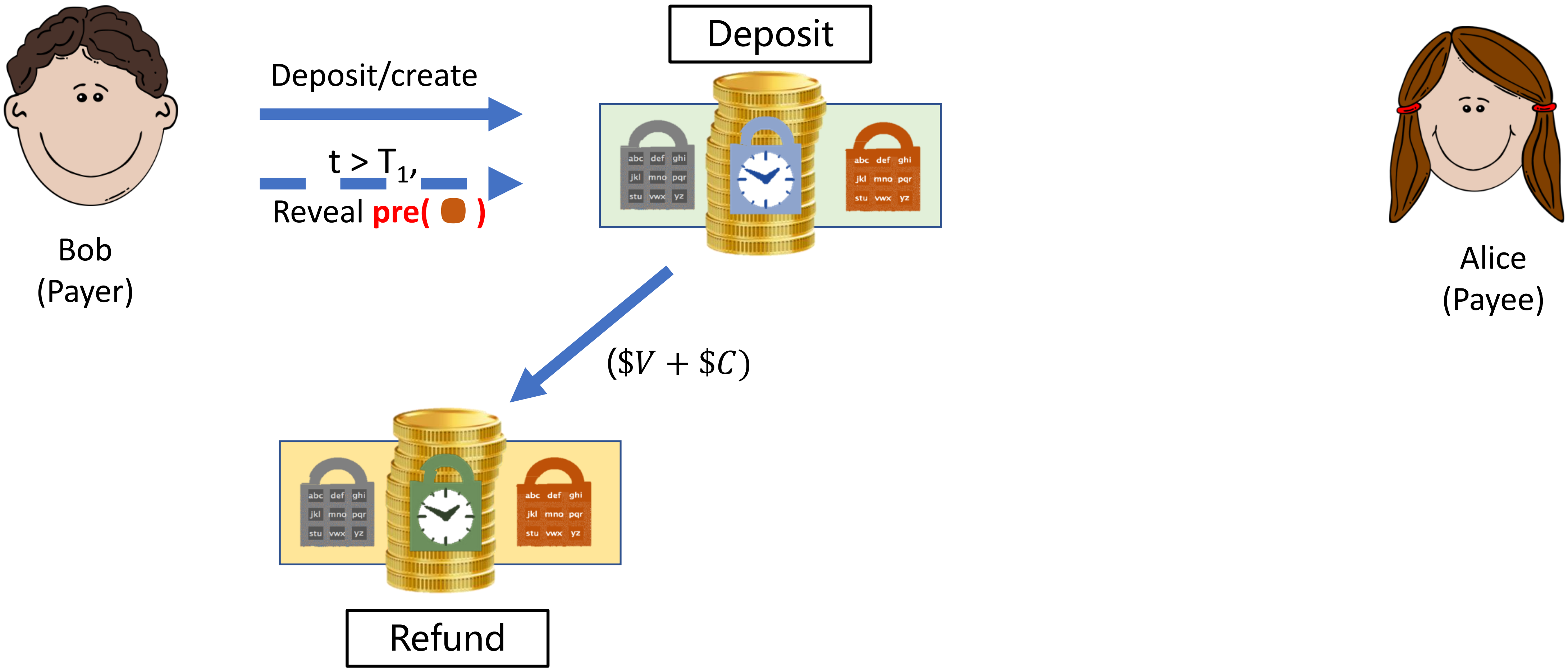
Deposit

Reveal pre(■)

(V)

Alice
(Payee)

He-HTLC: An Incentive Compatible HTLC



Bob
(Payer)

Deposit/create

$t > T_1$,
Reveal pre(●)

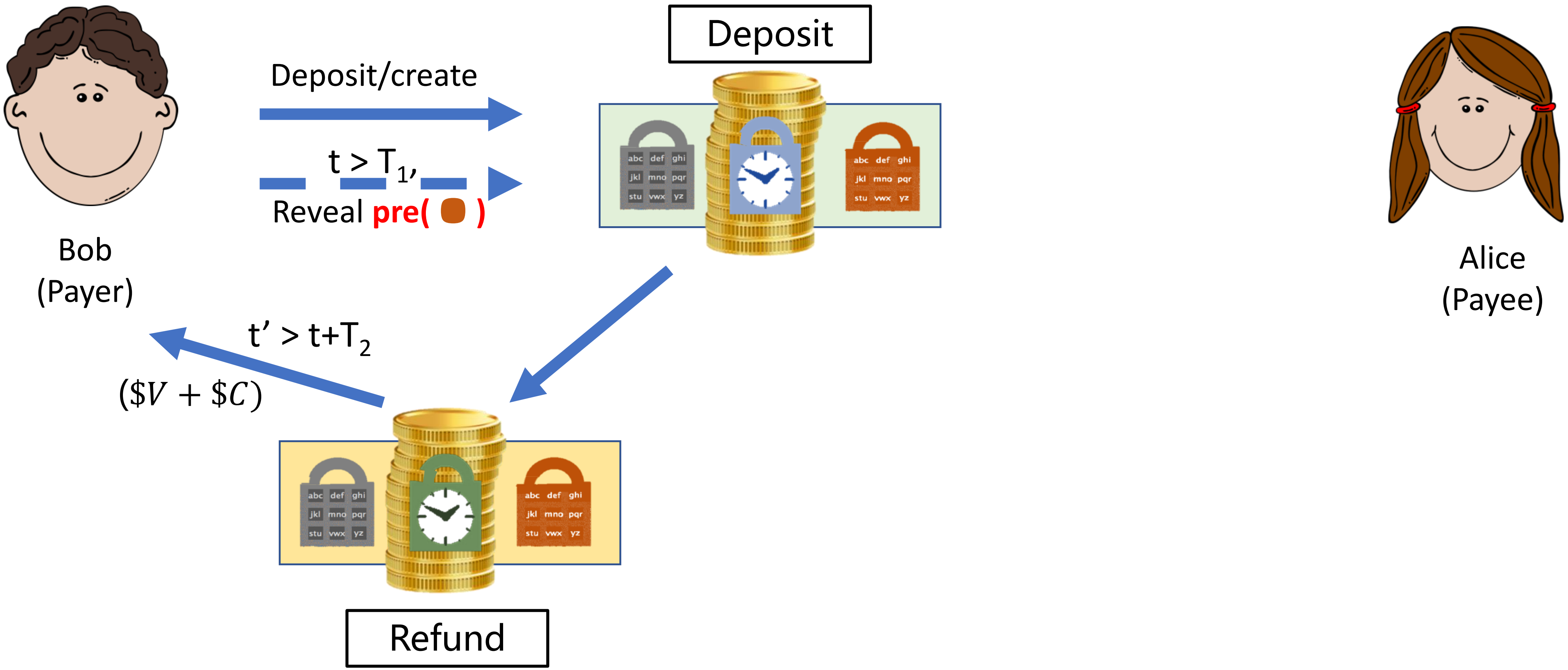
Deposit

($\$V + \$C$)

Refund

Alice
(Payee)

He-HTLC: An Incentive Compatible HTLC



Bob
(Payer)

Deposit/create

$t > T_1$,
Reveal **pre(●)**

Deposit

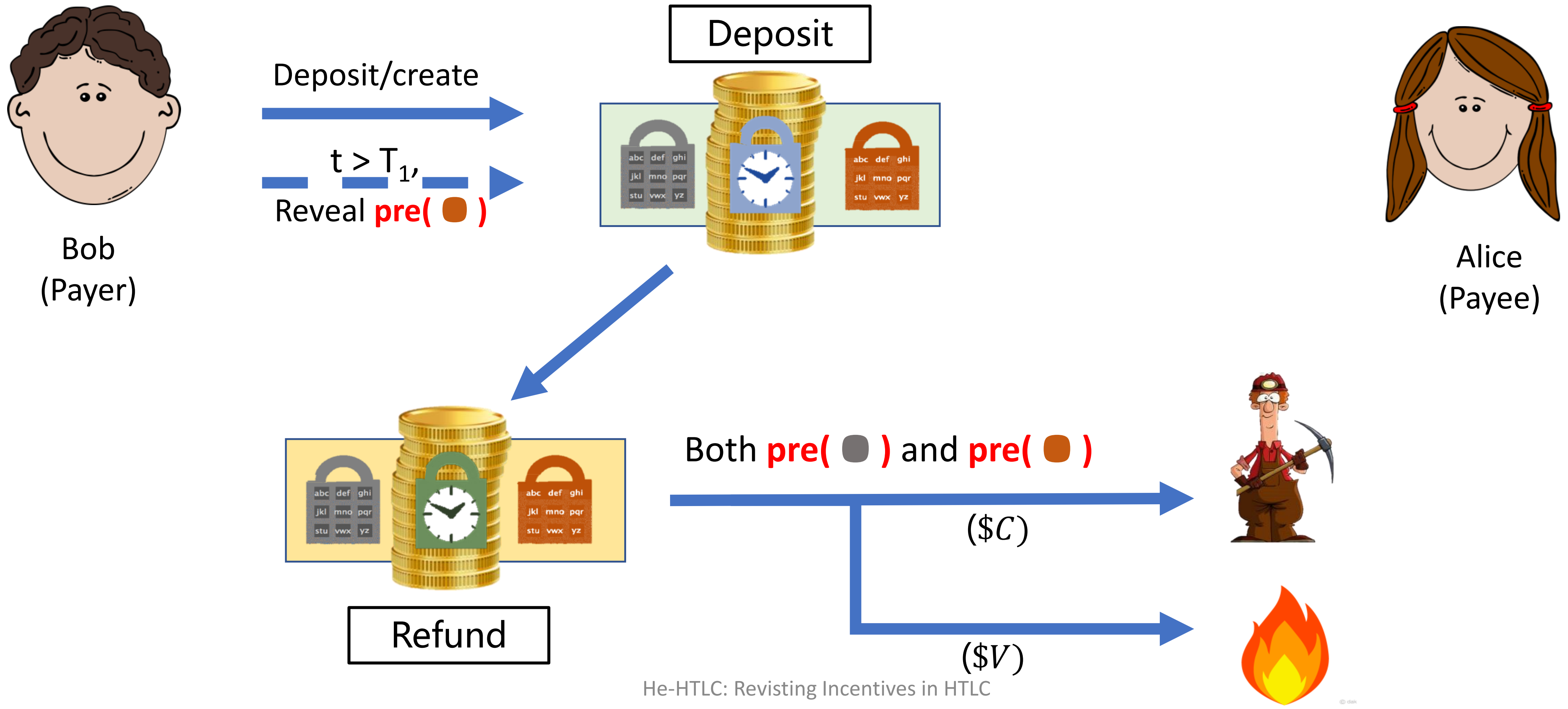
Alice
(Payee)

$t' > t+T_2$

($\$V + \$C$)

Refund

He-HTLC: An Incentive Compatible HTLC



Bob
(Payer)

Deposit/create

$t > T_1$,
Reveal **pre(🟠)**

Deposit

Alice
(Payee)

Refund

Both **pre(●)** and **pre(🟠)**

($\$C$)

($\$V$)

He-HTLC: An Incentive Compatible HTLC

✓ No incentive-based attacks on HTLCs even with 100% active miners!

He-HTLC: An Incentive Compatible HTLC

✓ No incentive-based attacks on HTLCs even with 100% active miners!

✓ Low and user adjustable collateral ($\$C < \$V$)

He-HTLC: An Incentive Compatible HTLC

✓ No incentive-based attacks on HTLCs even with 100% active miners!

✓ Low and user adjustable collateral ($\$C < \$V$)

✓ A lightweight Bitcoin implementation (no new op-codes)

He-HTLC: Revisiting Incentives in HTLC

Sarisht Wadhwa[§]
Duke University
sarisht.wadhwa@duke.edu

Jannis Stöter[§]
Duke University
jannis.stoeter@alumni.duke.edu

Fan Zhang
Duke University
fan.zhang@duke.edu

Kartik Nayak
Duke University
kartik@cs.duke.edu

Thank You!

Contact : sarisht.wadhwa@duke.edu