

# One-Time Programs from Commodity Hardware

---

Harry Eldridge Aarushi Goel Matthew Green Abhishek Jain Maximilian Zinkus



# One-Time Programs [GKR08]

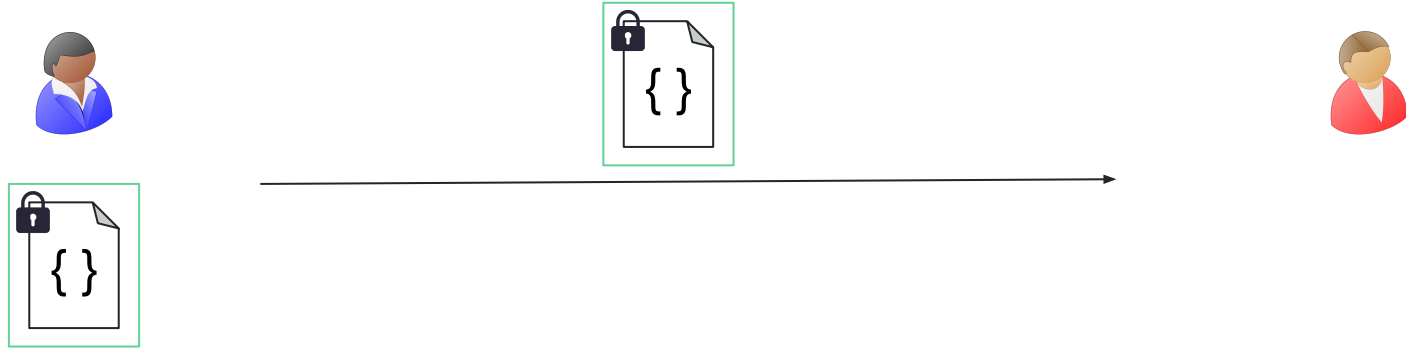
# One-Time Programs [GKR08]



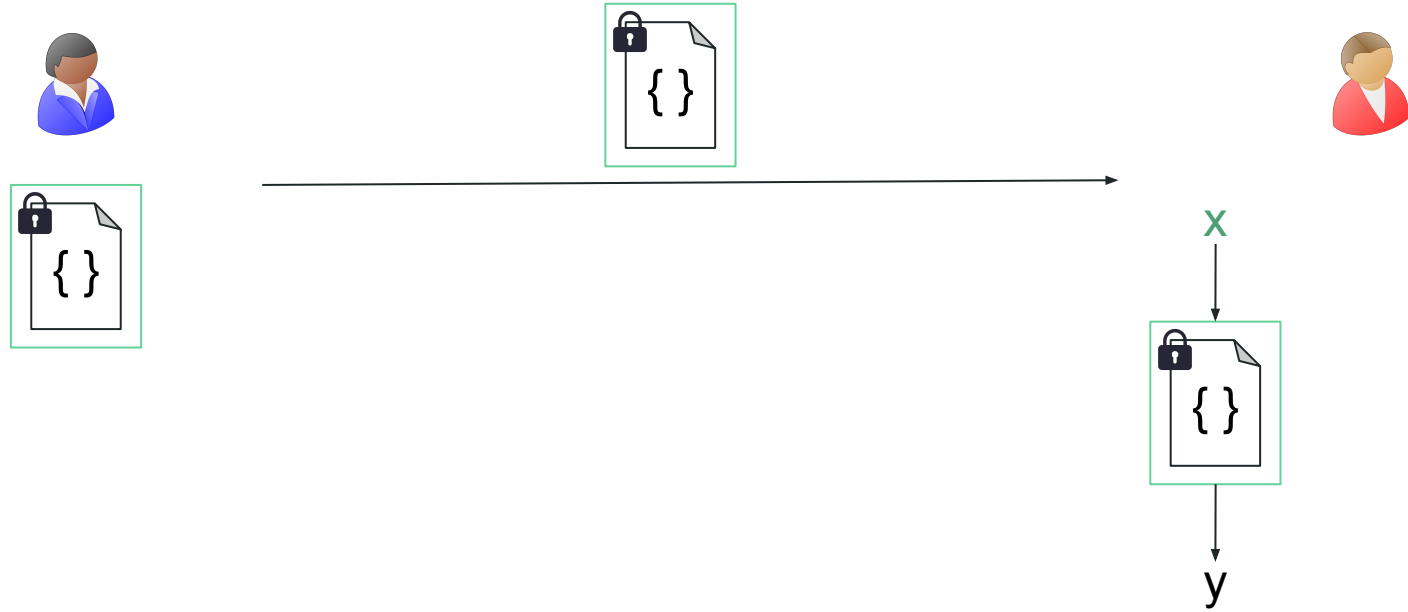
# One-Time Programs [GKR08]



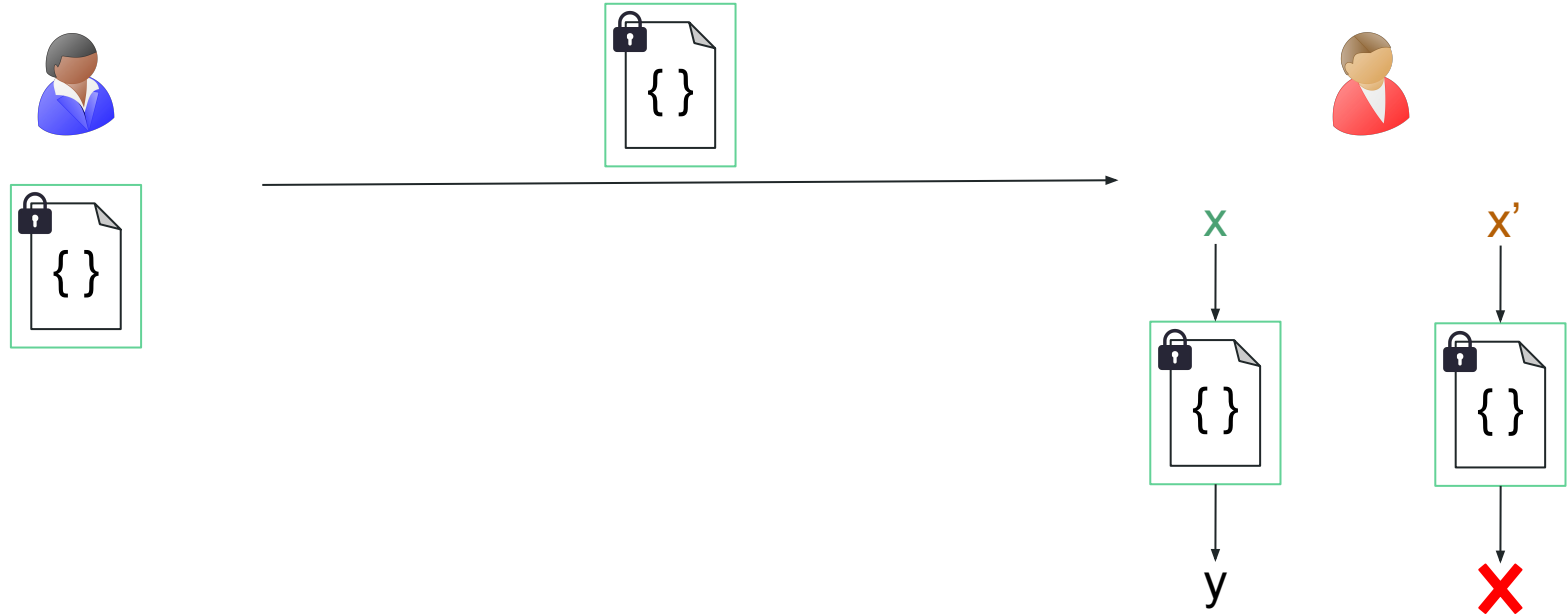
# One-Time Programs [GKR08]



# One-Time Programs [GKR08]



# One-Time Programs [GKR08]



# Applications

Limited attempt authentication



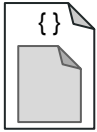
# Applications

Limited attempt authentication



# Applications

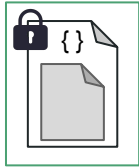
Limited attempt authentication



```
If input == 1234:  
    output file
```

# Applications

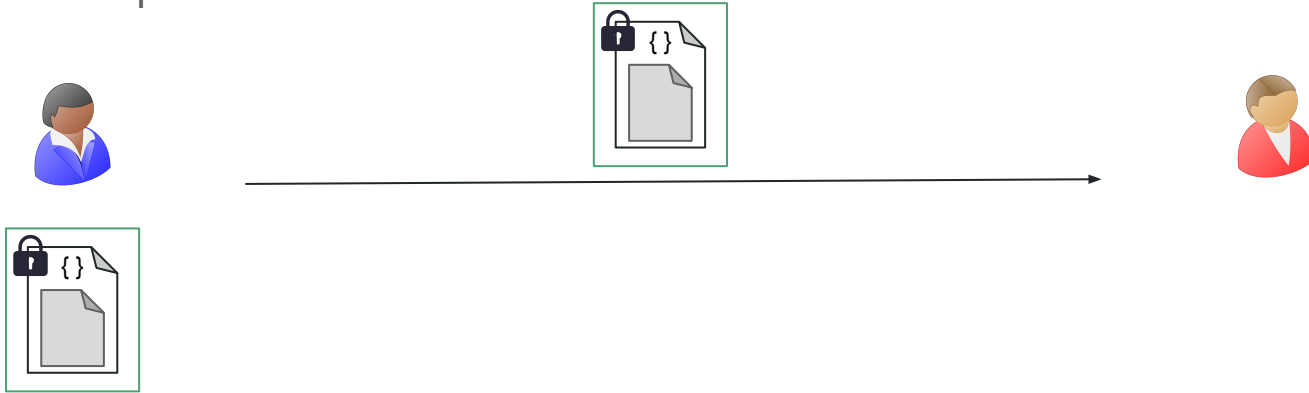
Limited attempt authentication



```
If input == 1234:  
    output file
```

# Applications

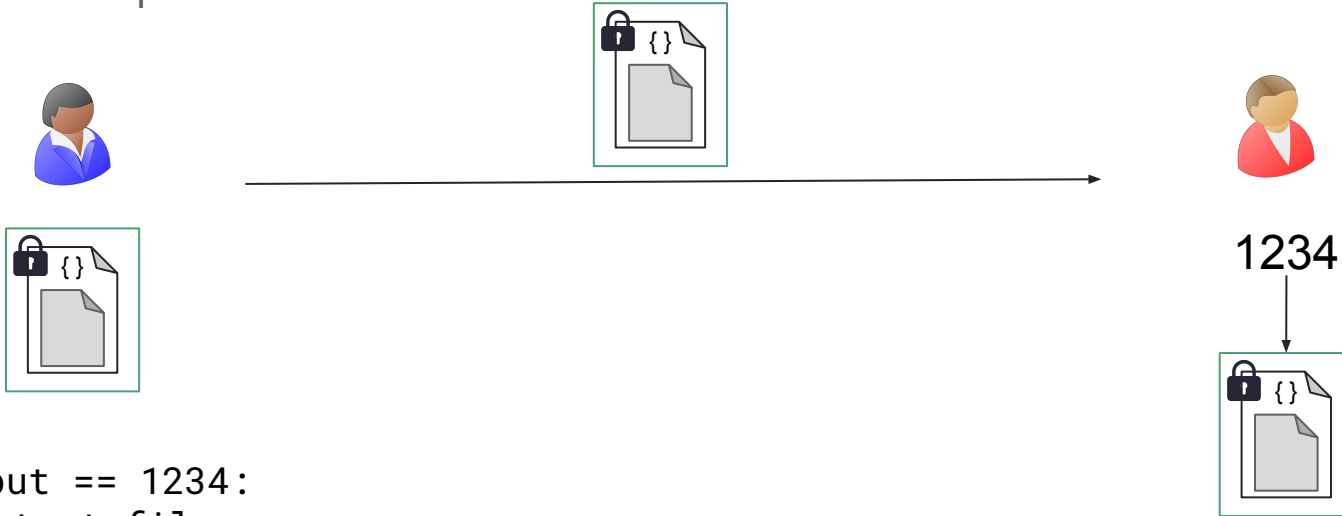
Limited attempt authentication



```
If input == 1234:  
    output file
```

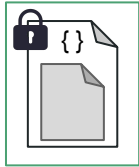
# Applications

Limited attempt authentication



# Applications

## Limited attempt authentication



1234



If input == 1234:  
output file

# Other Applications

- Differentially-private data analysis
- Autonomous Ransomware

# Feasibility

- Cannot be realized only in software [GKR08]



# Feasibility

- Cannot be realized only in software [GKR08]



a b

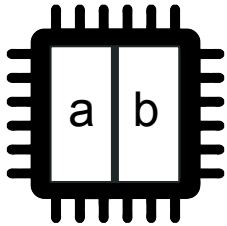


# Feasibility

- Cannot be realized only in software [GKR08]

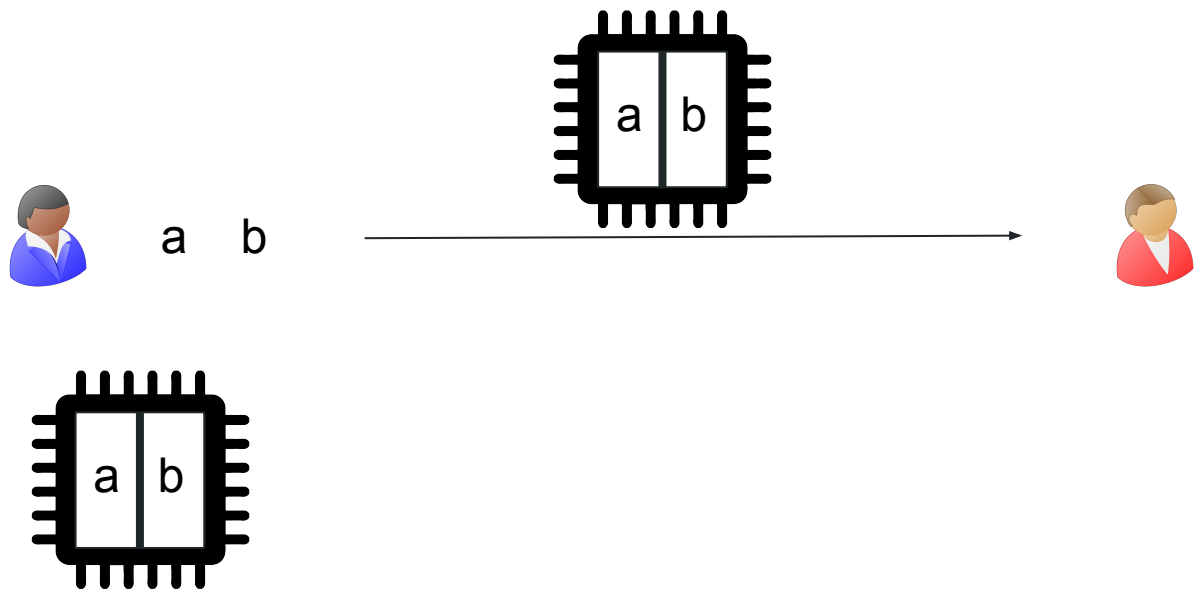


a b



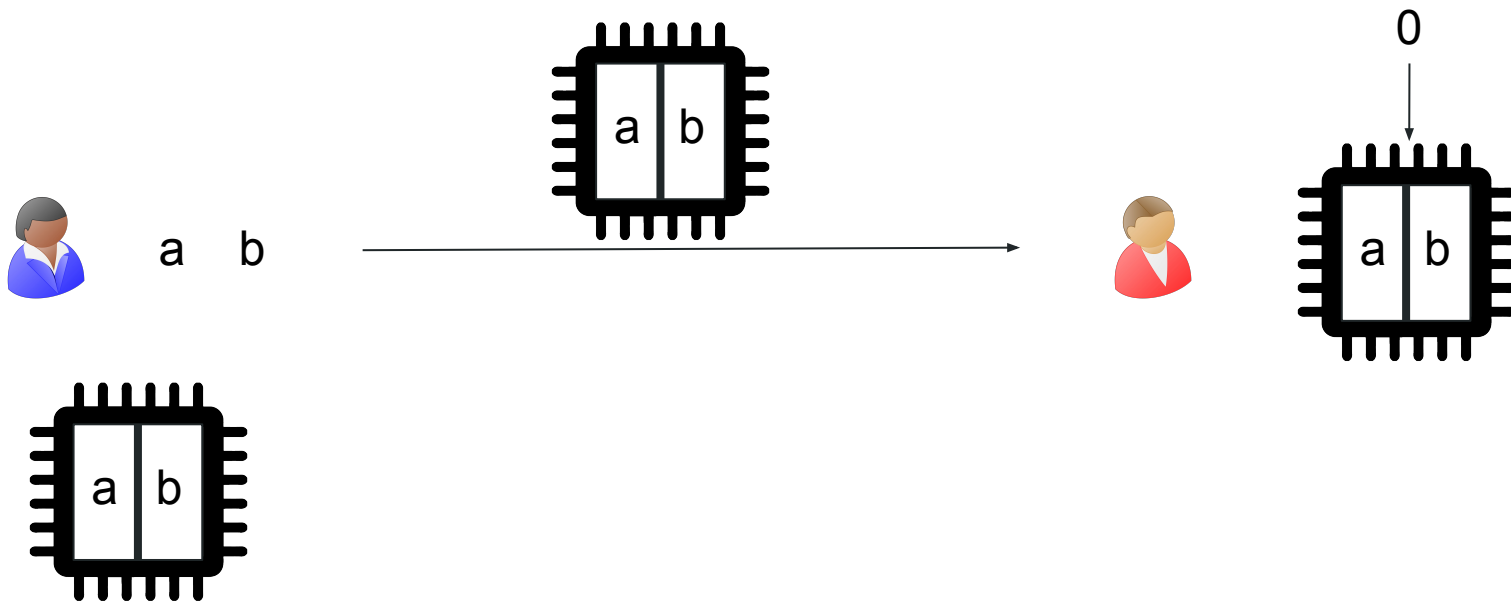
# Feasibility

- Cannot be realized only in software [GKR08]



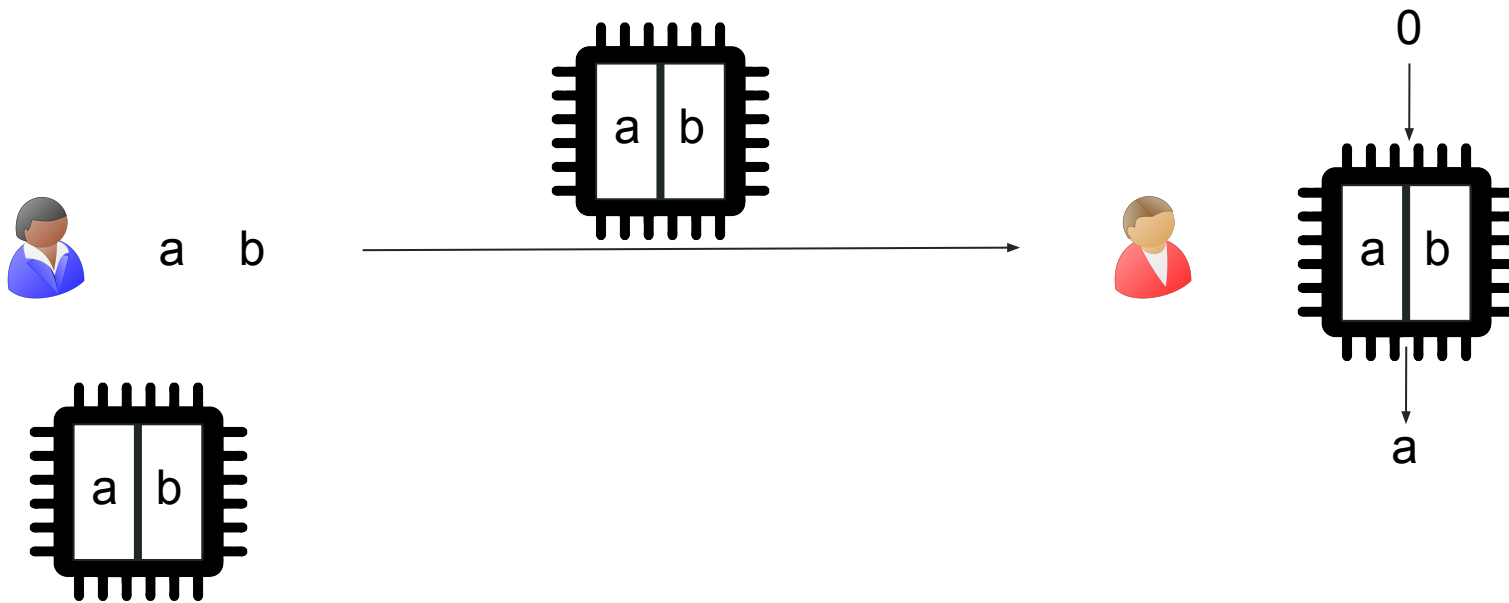
# Feasibility

- Cannot be realized only in software [GKR08]



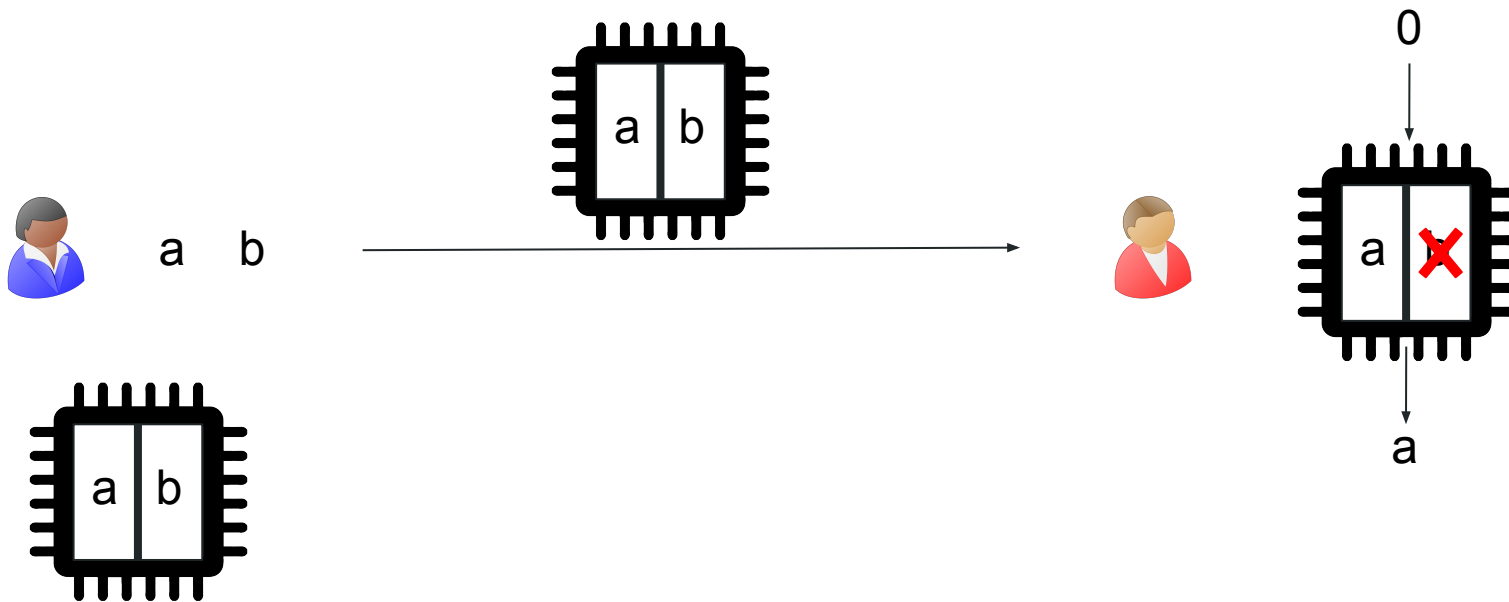
# Feasibility

- Cannot be realized only in software [GKR08]



# Feasibility

- Cannot be realized only in software [GKR08]



# Feasibility

- No known deployments of one-time programs
- One-time memory is not a standard offering

## Feasibility

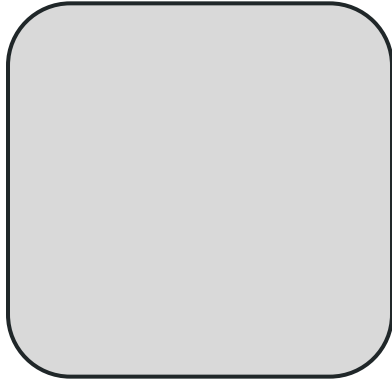
- No known deployments of one-time programs
- One-time memory is not a standard offering

**Can we build one-time programs from hardware that is already widely available?**



# Counter Lockboxes

# Counter Lockboxes



# Counter Lockboxes



S: abcd

# Counter Lockboxes

S: abcd

P: 1234

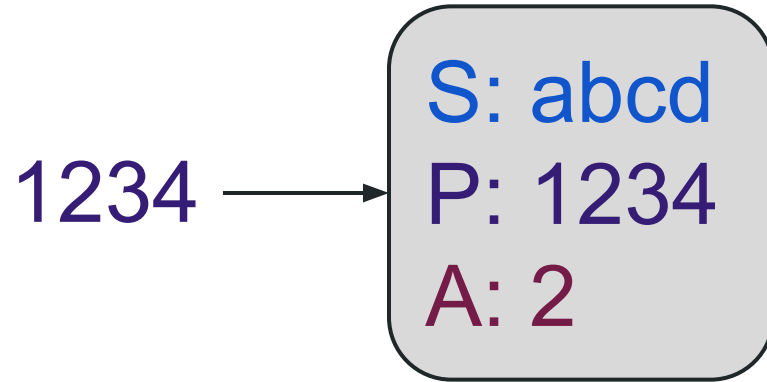
# Counter Lockboxes

S: abcd

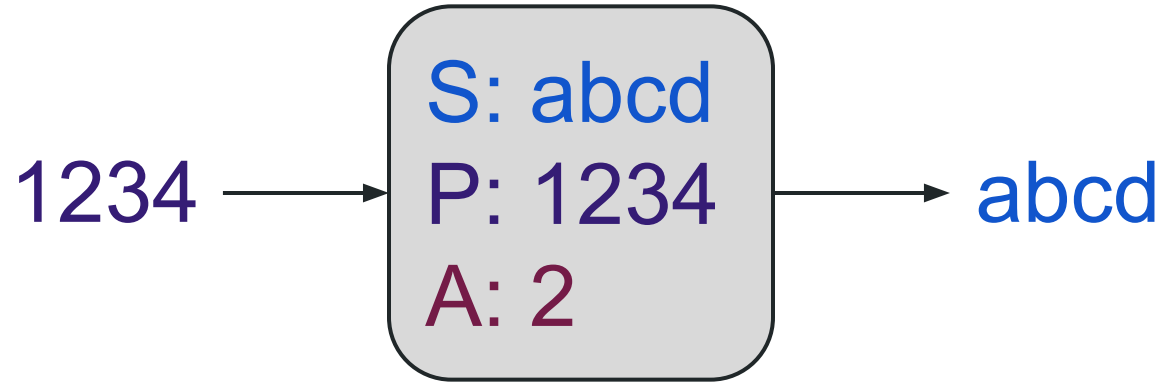
P: 1234

A: 2

# Counter Lockboxes



# Counter Lockboxes



# Counter Lockboxes

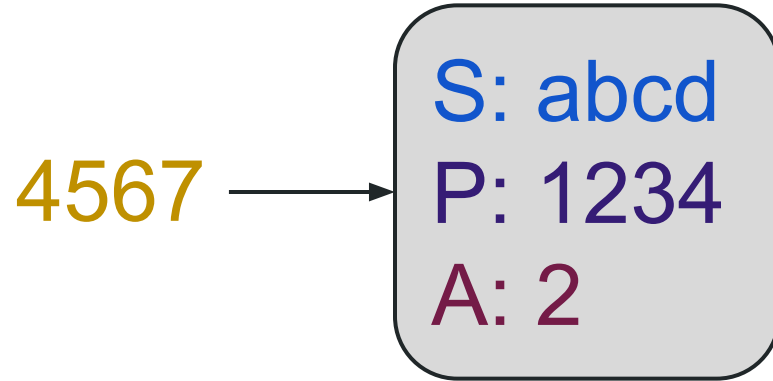
S: abcd

P: 1234

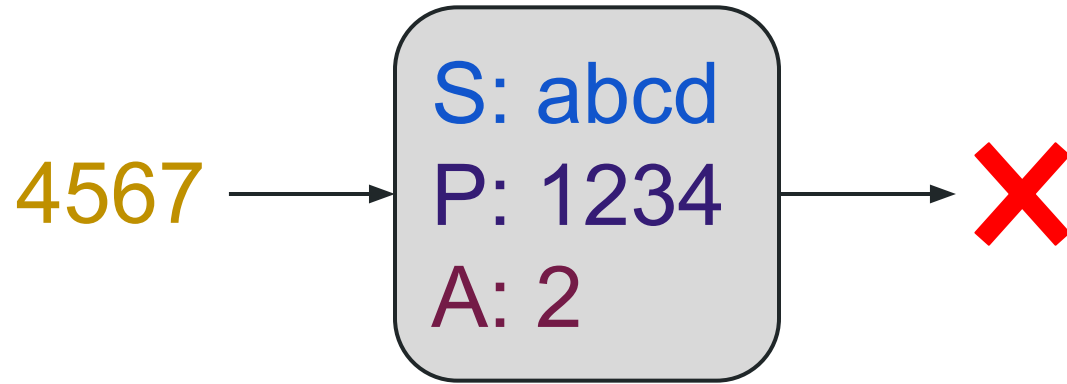
A: 2



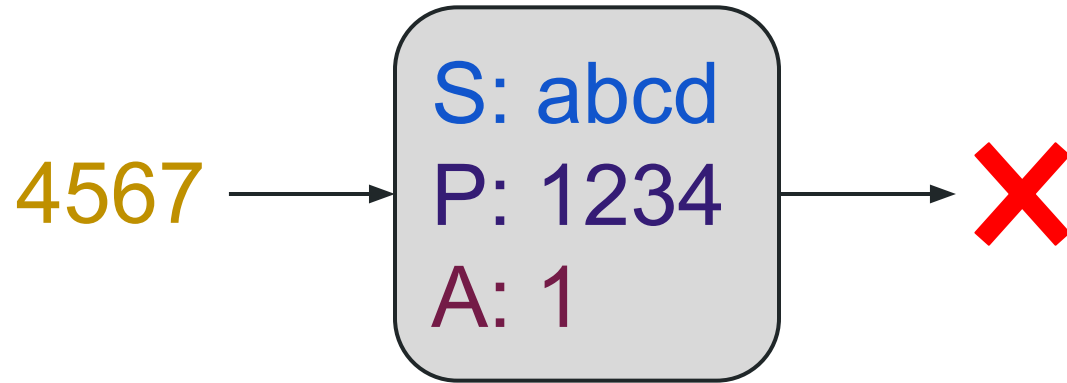
# Counter Lockboxes



# Counter Lockboxes



# Counter Lockboxes



# Counter Lockboxes

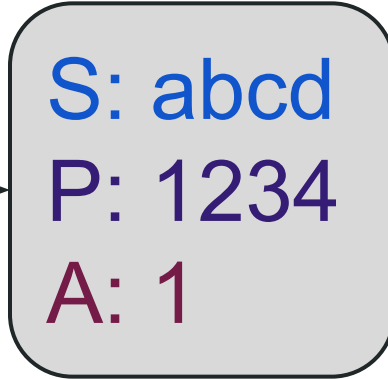
S: abcd

P: 1234

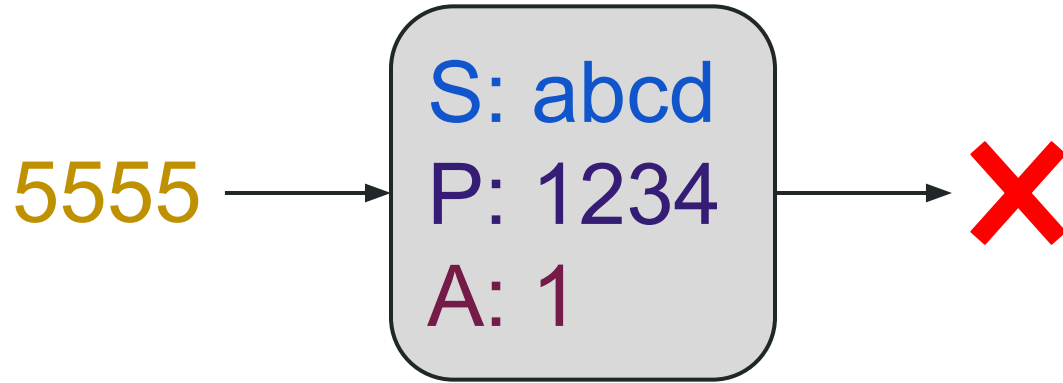
A: 1

# Counter Lockboxes

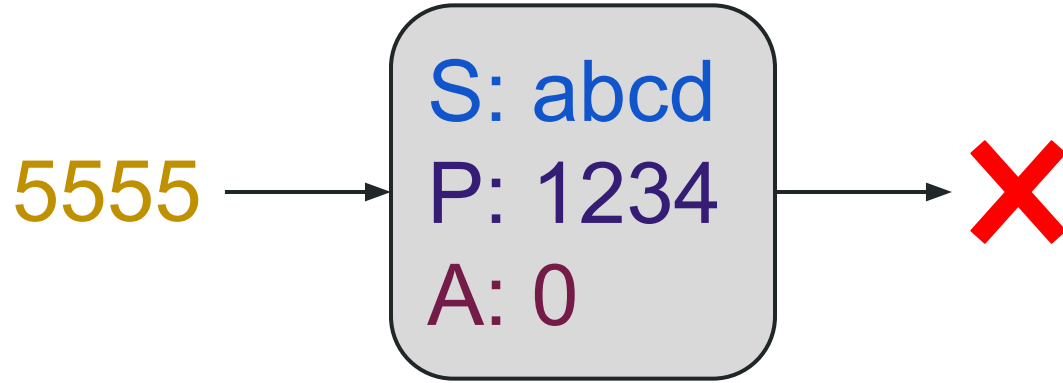
5555



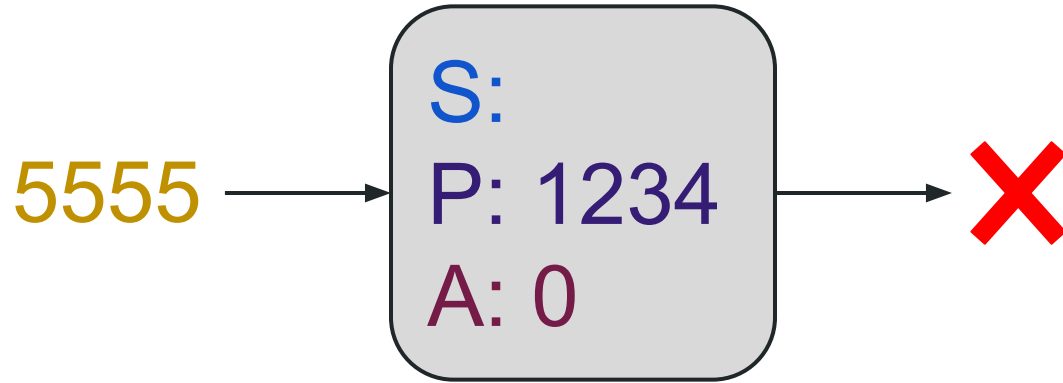
# Counter Lockboxes



# Counter Lockboxes



# Counter Lockboxes





# Counter Lockboxes

- Secure co-processors in smartphones
  - Apple Secure Enclave Processor
  - Google Titan M2
- Secure hardware used by data backup services
  - Apple Cloud Key Vault
  - Android Backup
  - WhatsApp backup
  - Signal Secure Value Recovery

# Results

## Results

Assuming the existence of one-way functions, there exist one-time programs for general circuits using  $O(1)$  lockboxes per input bit

## Results

Assuming the existence of one-way functions, there exist one-time programs for general circuits using  $O(1)$  lockboxes per input bit



Using stronger assumptions (malicious receiver laconic OT), there exist one-time programs for general circuits using  $O(\lambda)$  *total* lockboxes

# Our Approach



# Our Approach

- CLBs  “leaky” one-time memory

# Our Approach

- CLBs  “leaky” one-time memory
- OT combiner + leaky OTM  one-time programs

# Our Approach

- CLBs  “leaky” one-time memory
- OT combiner + leaky OTM  one-time programs
- Robust garbling [ABH+21] to reduce the number of lockboxes



# Leaky One-time Memory from Counter Lockboxes

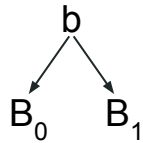
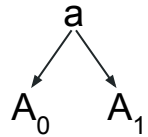


a

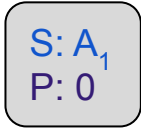
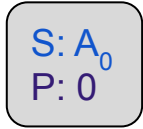
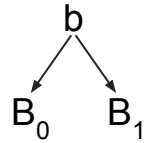
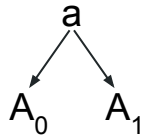
b



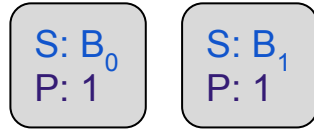
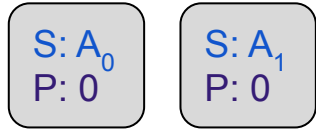
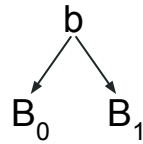
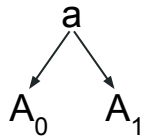
# Leaky One-time Memory from Counter Lockboxes



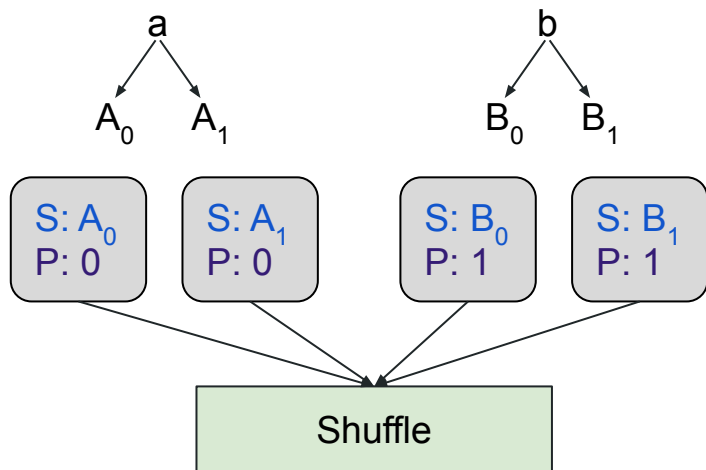
# Leaky One-time Memory from Counter Lockboxes



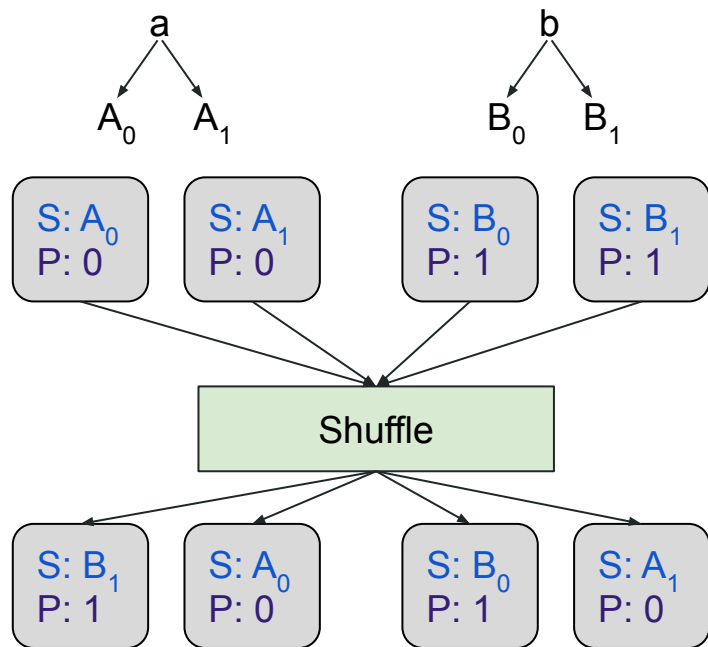
# Leaky One-time Memory from Counter Lockboxes



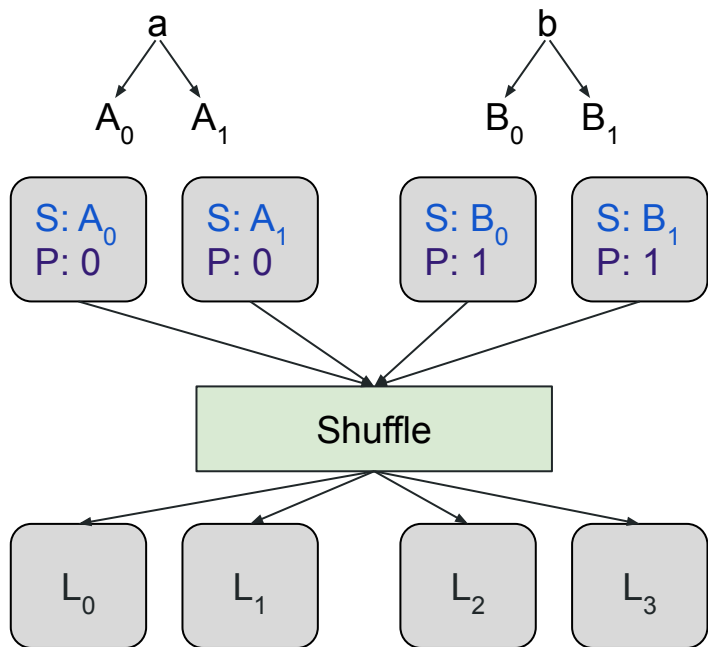
# Leaky One-time Memory from Counter Lockboxes



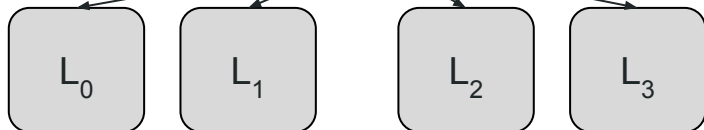
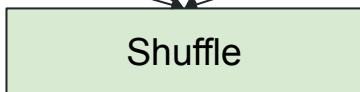
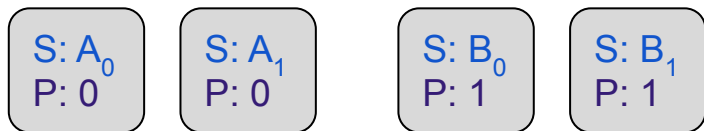
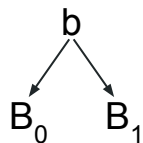
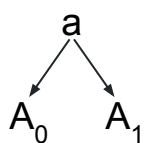
# Leaky One-time Memory from Counter Lockboxes



# Leaky One-time Memory from Counter Lockboxes

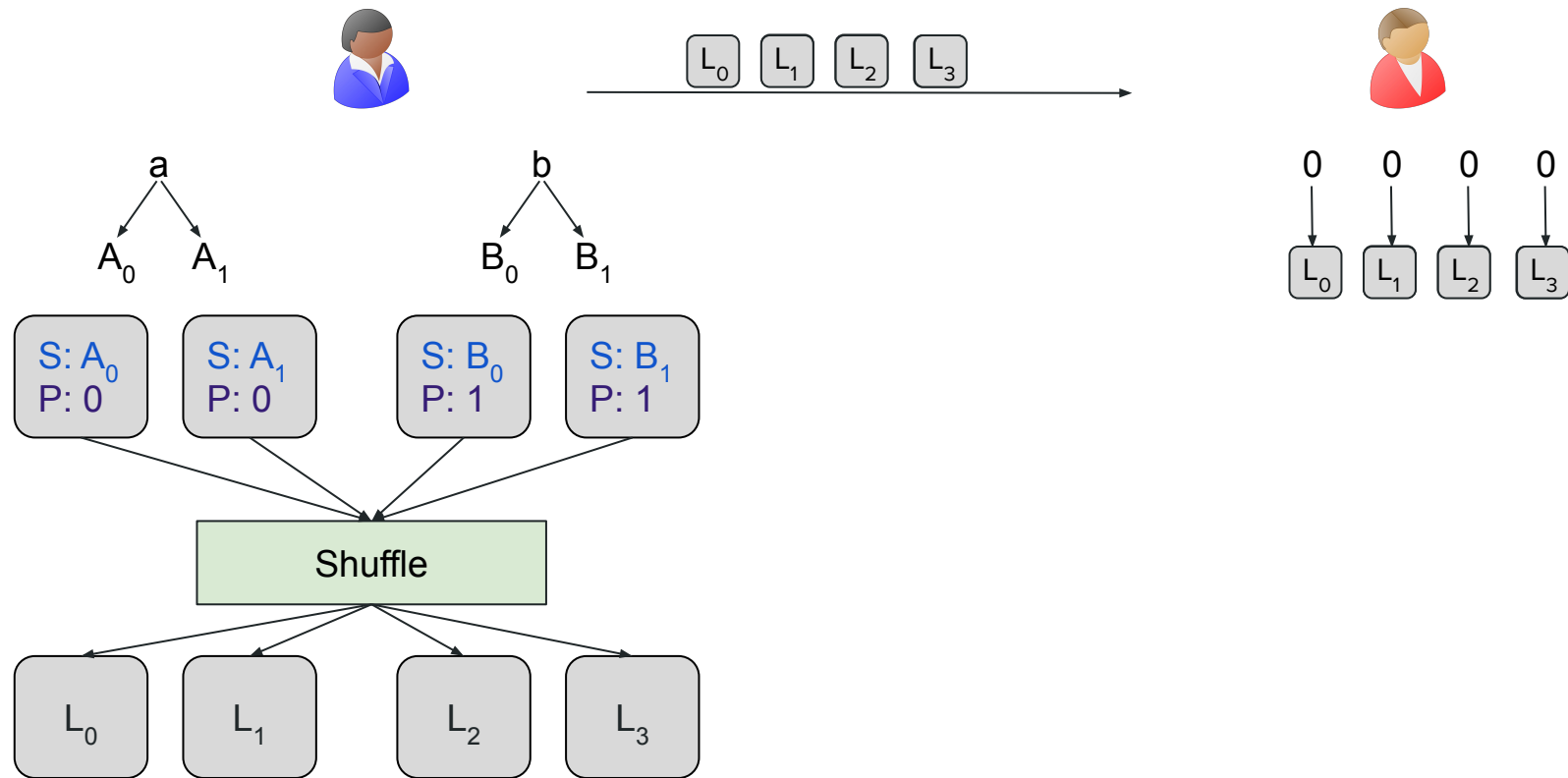


# Leaky One-time Memory from Counter Lockboxes

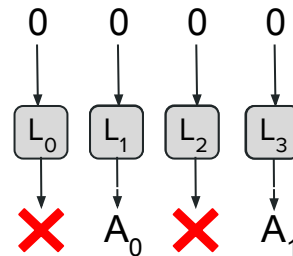
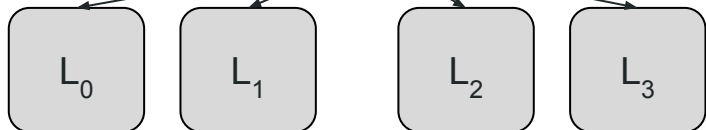
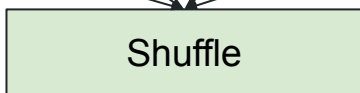
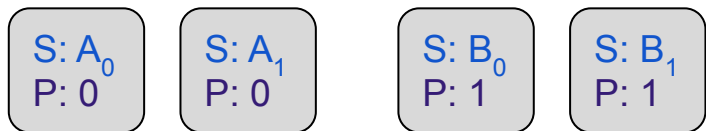
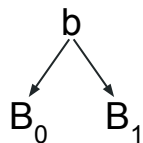
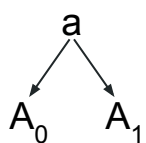




# Leaky One-time Memory from Counter Lockboxes



# Leaky One-time Memory from Counter Lockboxes



$$A_0 + A_1 = a$$

# Conclusions and Future Work

- **Result:** One-time programs are possible using widely available secure hardware
- **Open Question:** Can the number of lockboxes used be reduced?

## Conclusions and Future Work

- **Result:** One-time programs are possible using widely available secure hardware
- **Open Question:** Can the number of lockboxes used be reduced?

Thank you!