

# The Phantom Tollbooth: Privacy-Preserving Toll Collection in the Presence of Driver Collusion

---

**Sarah Meiklejohn (UC San Diego)**

Keaton Mowery (UC San Diego)

Stephen Checkoway (UC San Diego)

Hovav Shacham (UC San Diego)

# Motivation: how tolling works today

---

# Motivation: how tolling works today

---



# Motivation: how tolling works today





# Motivation: how tolling works today



This process leaves a lot to be desired in terms of **flexibility**:

# Motivation: how tolling works today



This process leaves a lot to be desired in terms of **flexibility**:

- How do we charge more according to the **time of day**?



# Motivation: how tolling works today



This process leaves a lot to be desired in terms of **flexibility**:

- How do we charge more according to the **time of day**?
- Or as drivers enter **city centers**?

# Motivation: how tolling works today

---

# Motivation: how tolling works today

---





# Motivation: how tolling works today

---



# Motivation: how tolling works today

---



Core tension between privacy and desire for more flexible toll pricing



# Motivation: how tolling works today

---



Core tension between privacy and desire for more flexible toll pricing

- In this talk we'll see our system, **Milo**, which allows for **fine-grained pricing policies** without sacrificing drivers' **privacy**

# Motivation: how tolling works today

---



Core tension between privacy and desire for more flexible toll pricing

- In this talk we'll see our system, [Milo](#), which allows for **fine-grained pricing policies** without sacrificing drivers' **privacy**
- In the process, we strongly guarantee that drivers remain **honest**

Previous work [BKS05, BC06, TDKP07, dJJ08, ...]

---



# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

- **Fine-grained policy**: uses small road segments (where,when)

# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

- **Fine-grained policy**: uses small road segments (where,when)
- **Privacy**: uses Tor to maintain anonymity while driver uploads segments

# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

- **Fine-grained policy**: uses small road segments (where,when)
- **Privacy**: uses Tor to maintain anonymity while driver uploads segments
- **Honesty**: relies on audits wherein driver is asked to verify locations

# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

- **Fine-grained policy**: uses small road segments (where,when)
- **Privacy**: uses Tor to maintain anonymity while driver uploads segments
- **Honesty**: relies on audits wherein driver is asked to verify locations

USENIX Security 2010: **PrETP** [BRTPVG]

- **Fine-grained policy**: again uses small road segments
- **Privacy**: drivers commit to segments in a way that eliminates need for Tor
- **Honesty**: again relies on audits



# Previous work [BKS05,BC06,TDKP07,dJJ08,...]

---

USENIX Security 2009: **VPriv** [PBB]

- **Fine-grained policy**: uses small road segments (where,when)
- **Privacy**: uses Tor to maintain anonymity while driver uploads segments
- **Honesty**: relies on audits wherein driver is asked to verify locations

USENIX Security 2010: **PrETP** [BRTPVG]

- **Fine-grained policy**: again uses small road segments
- **Privacy**: drivers commit to segments in a way that eliminates need for Tor
- **Honesty**: again relies on audits

A potential problem: keeping colluding drivers honest

---

# A potential problem: keeping colluding drivers honest

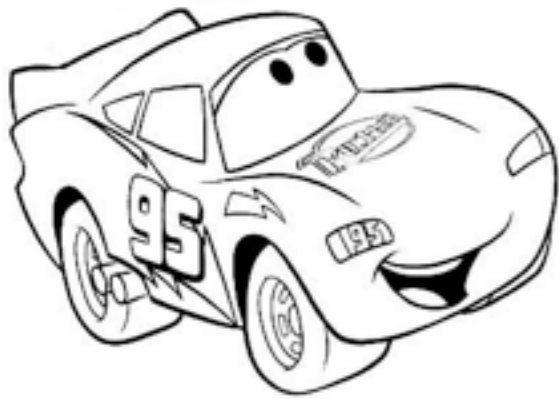
---

In these audits, we see a challenge/response behavior:

# A potential problem: keeping colluding drivers honest

---

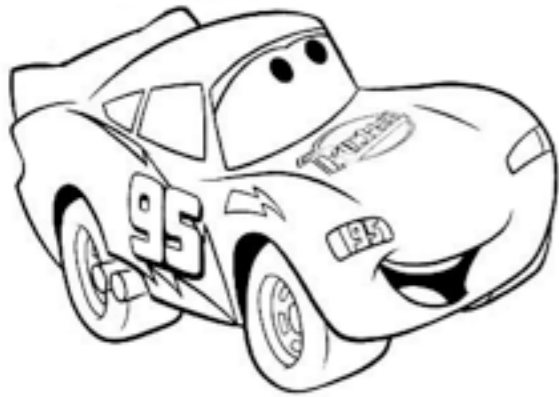
In these audits, we see a challenge/response behavior:



# A potential problem: keeping colluding drivers honest

---

In these audits, we see a challenge/response behavior:

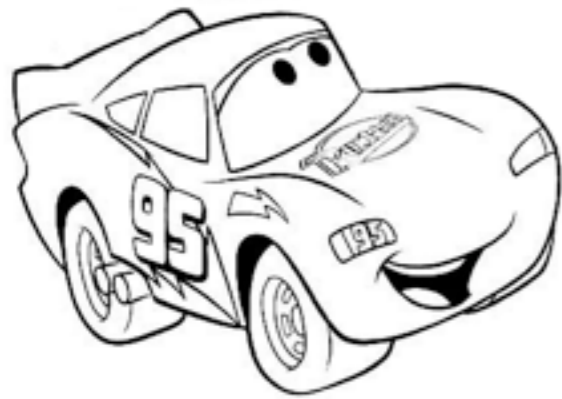




# A potential problem: keeping colluding drivers honest

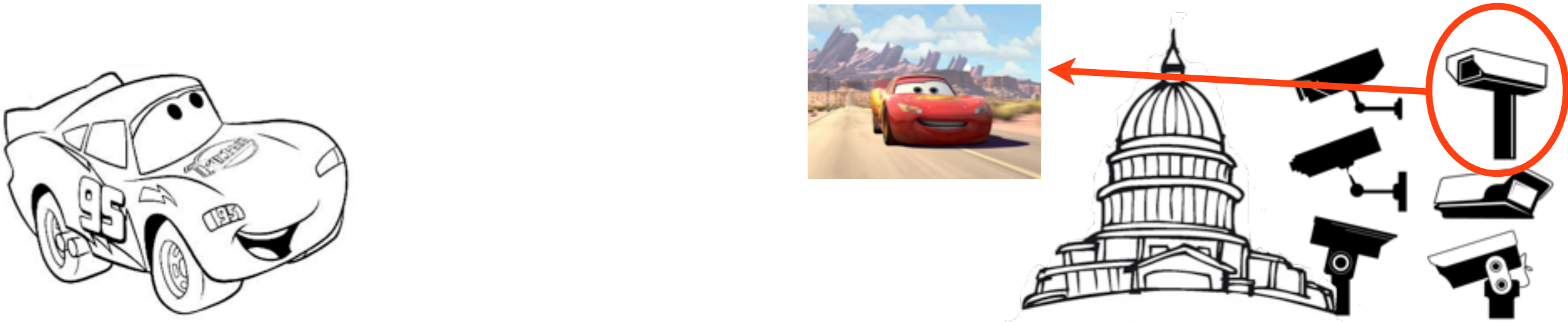
---

In these audits, we see a challenge/response behavior:



# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:



# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:



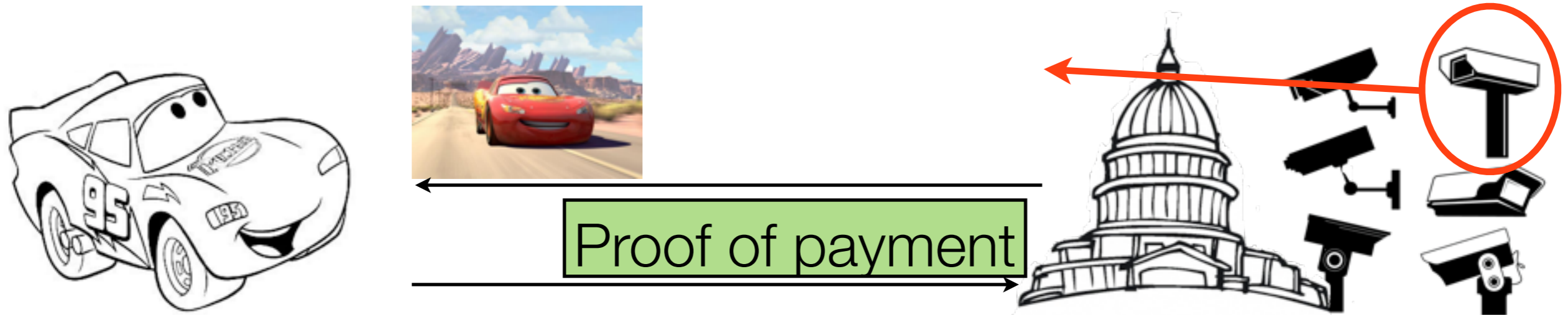
# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:



# A potential problem: keeping colluding drivers honest

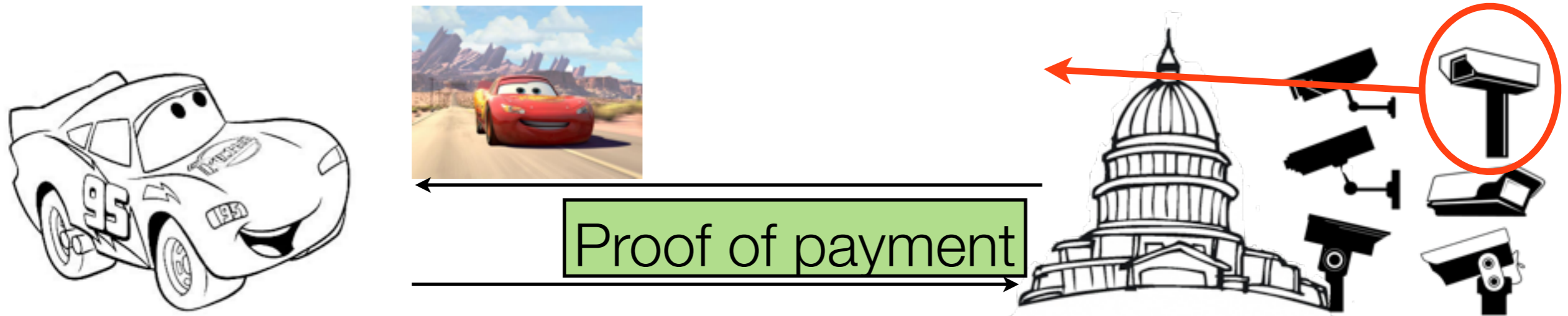
In these audits, we see a challenge/response behavior:





# A potential problem: keeping colluding drivers honest

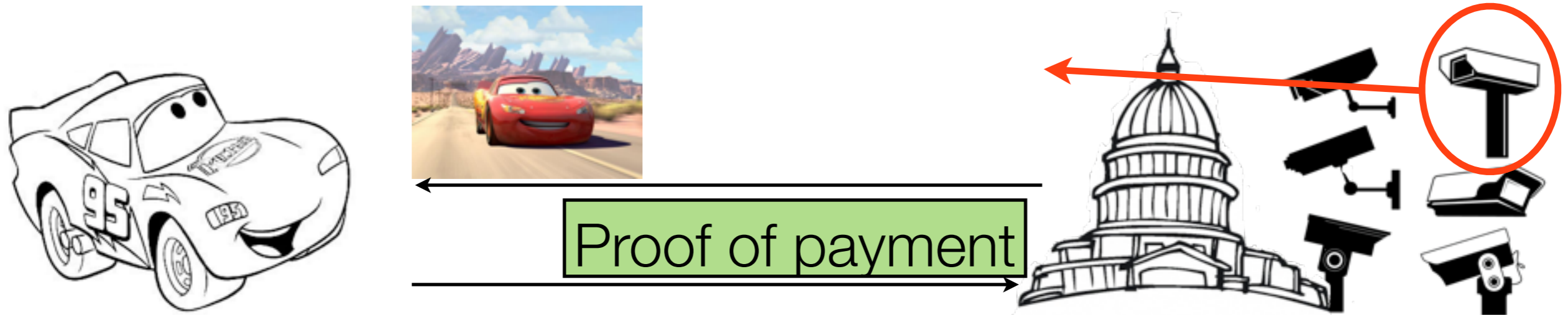
In these audits, we see a challenge/response behavior:



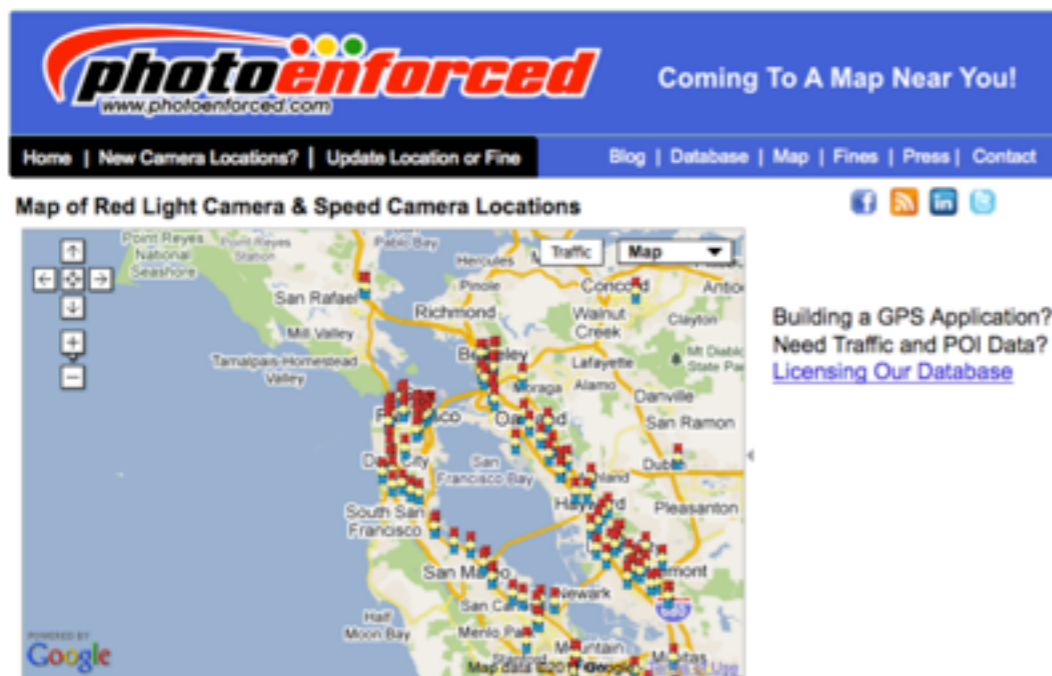
So the authority **reveals to the driver the segment in which he was seen!** This information can then be shared to help drivers avoid cameras in the future

# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:

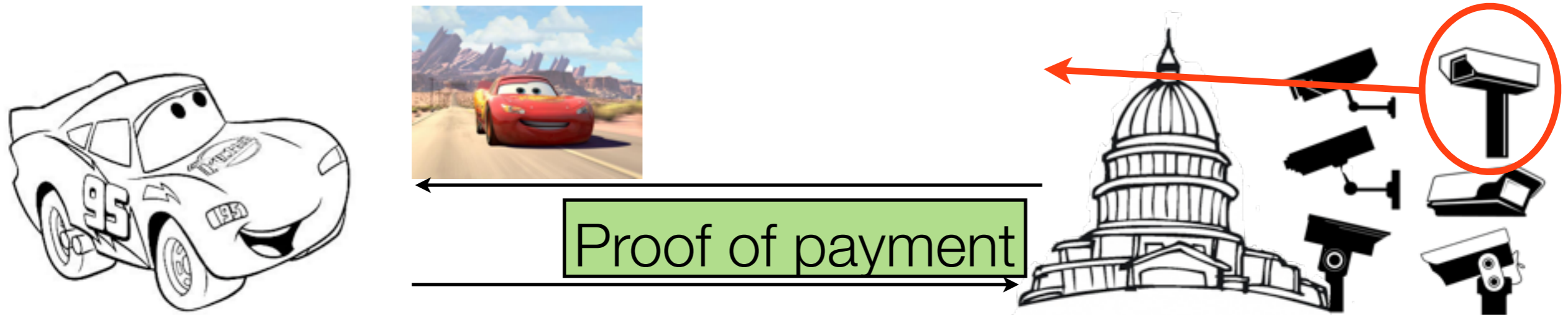


So the authority **reveals to the driver the segment in which he was seen!** This information can then be shared to help drivers avoid cameras in the future

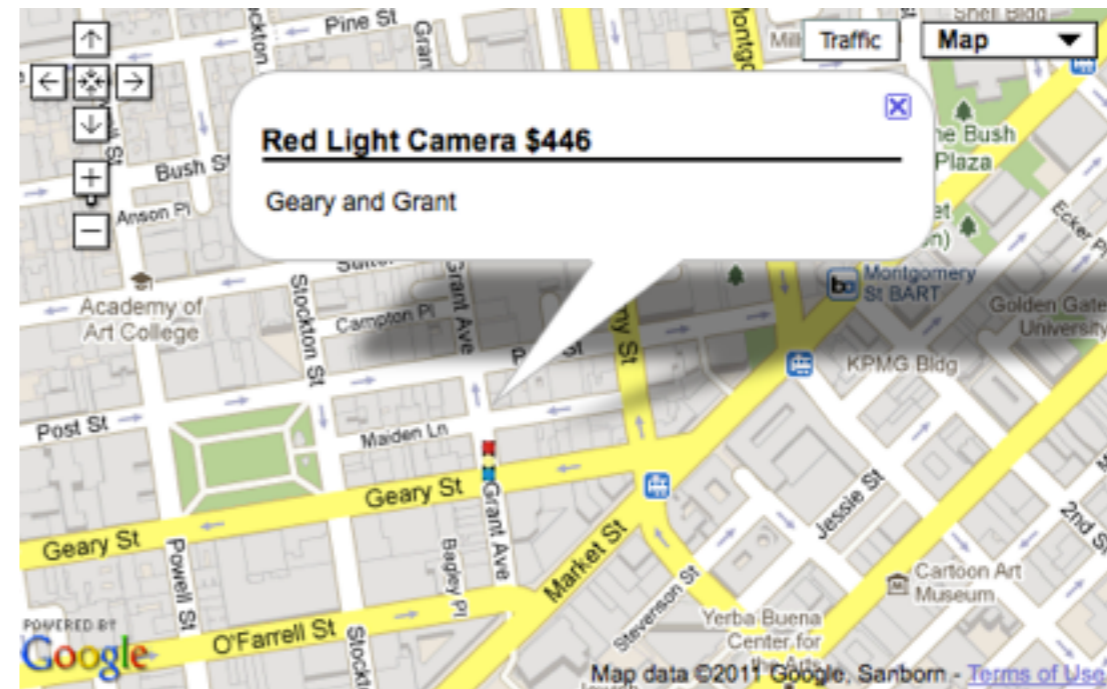
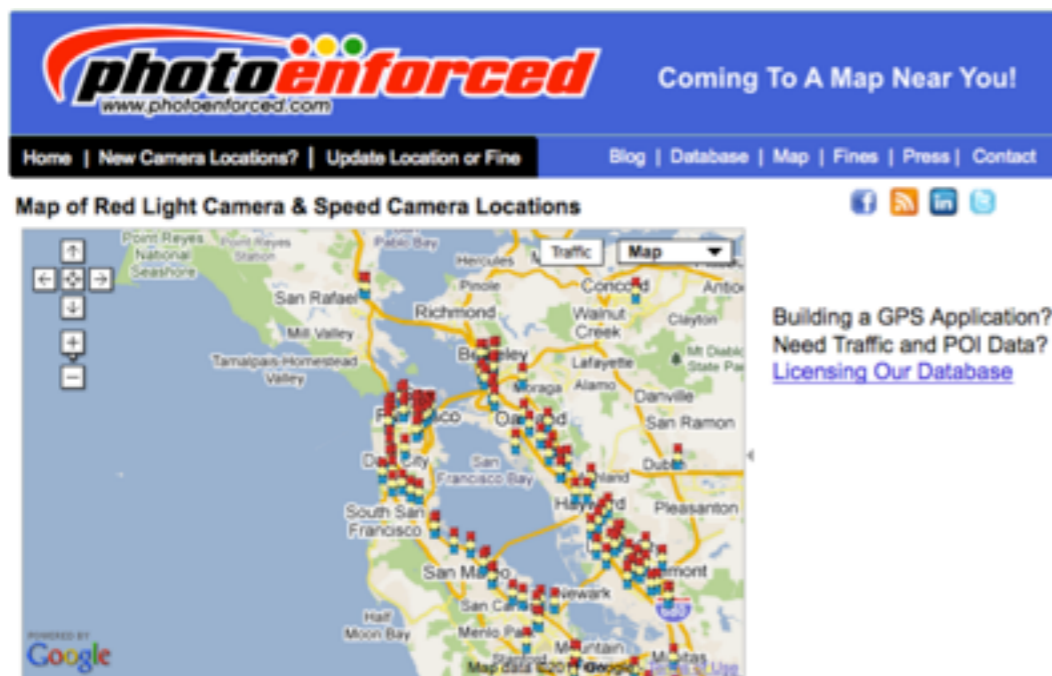


# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:



So the authority **reveals to the driver the segment in which he was seen!** This information can then be shared to help drivers avoid cameras in the future





# A potential problem: keeping colluding drivers honest

In these audits, we see a challenge/response behavior:



## USENIX Security 2011: Milo

- **Fine-grained policy:** uses same small road segments (where, when)
- **Privacy:** drivers commit to segments in a way similar to PrETP
- **Honesty:** audit protocol no longer reveals locations to drivers



Building a GPS Application?  
Need Traffic and POI Data?  
[Licensing Our Database](#)



# Outline

---



# Outline

---

Cryptographic background

# Outline

---

Cryptographic background

Milo

# Outline

---

Cryptographic background

Milo

Evaluation

# Outline

---

Cryptographic background

Milo

Evaluation

Conclusions

# Outline

---

## Cryptographic background

Commitment schemes

Zero-knowledge proofs

Blind identity-based encryption

Milo

Evaluation

Conclusions



# Commitments [BCC88,P91]

---

# Commitments [BCC88,P91]

---



# Commitments [BCC88,P91]

---

My favorite  
number is 42



# Commitments [BCC88,P91]

---

My favorite  
number is 42



42



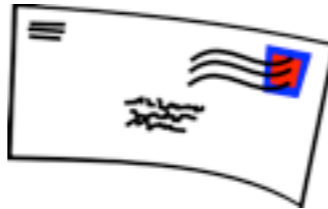
# Commitments [BCC88,P91]

---

My favorite  
number is 42



$c =$



42

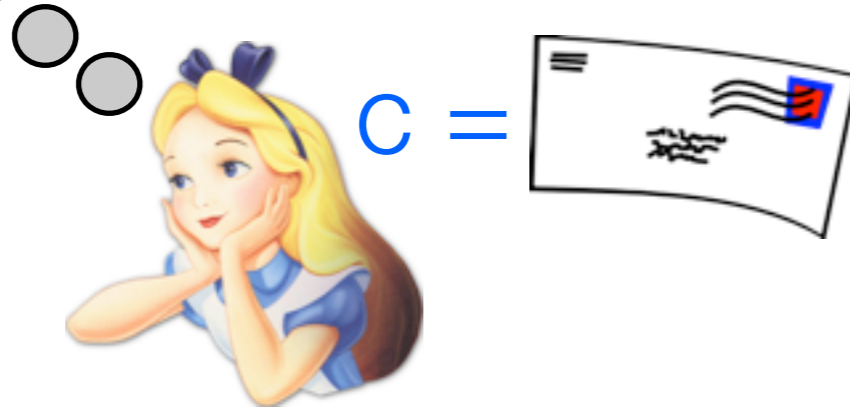




# Commitments [BCC88,P91]

---

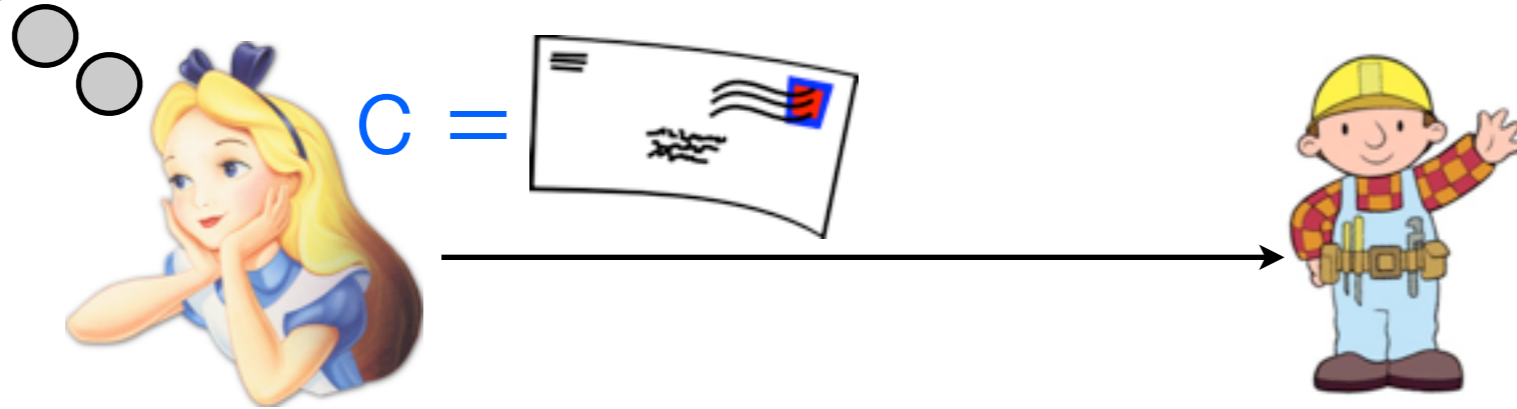
My favorite  
number is 42



# Commitments [BCC88,P91]

---

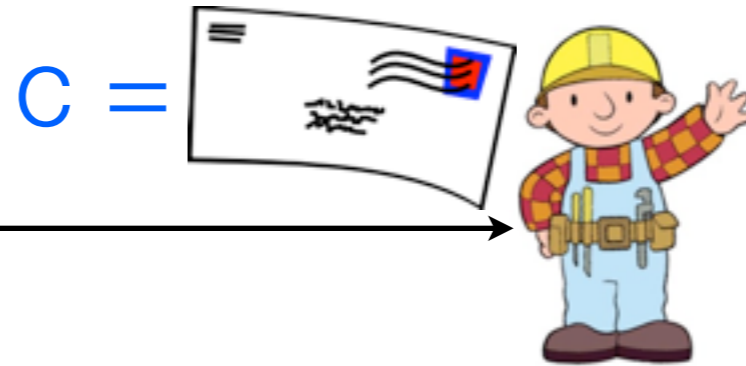
My favorite  
number is 42



# Commitments [BCC88,P91]

---

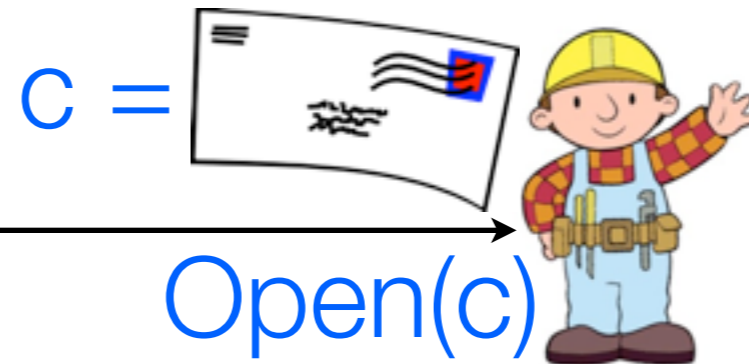
My favorite  
number is 42



# Commitments [BCC88,P91]

---

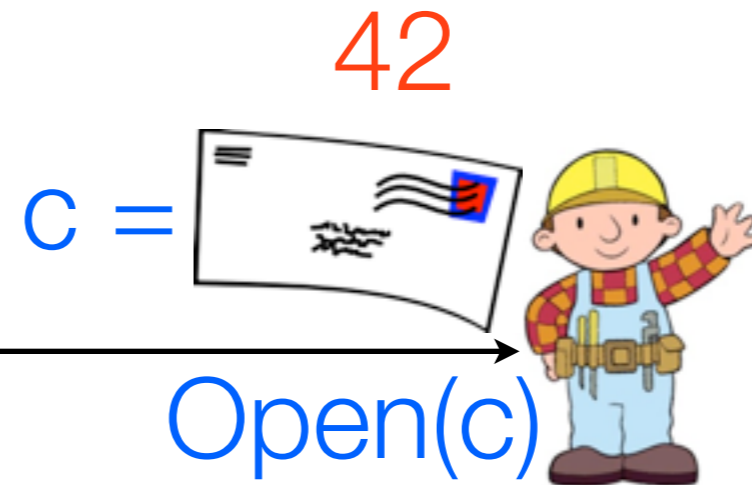
My favorite  
number is 42



# Commitments [BCC88,P91]

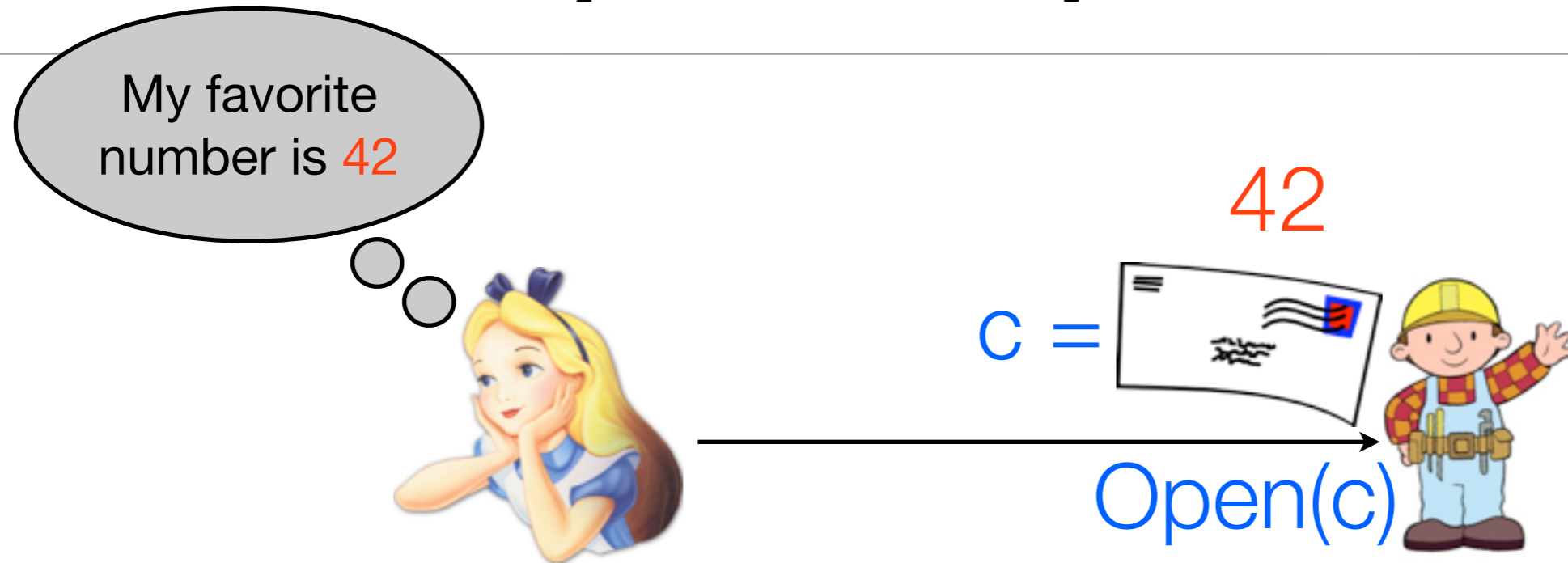
---

My favorite number is 42



# Commitments [BCC88,P91]

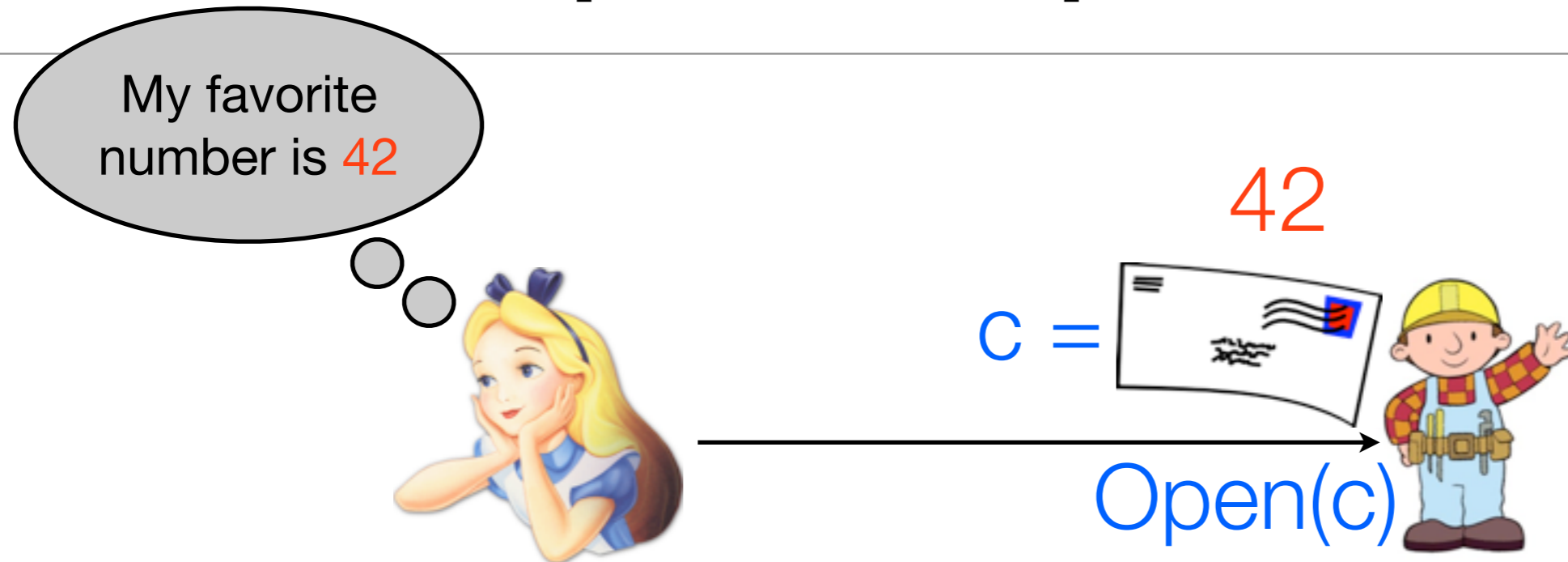
---



There are two important properties of commitments:



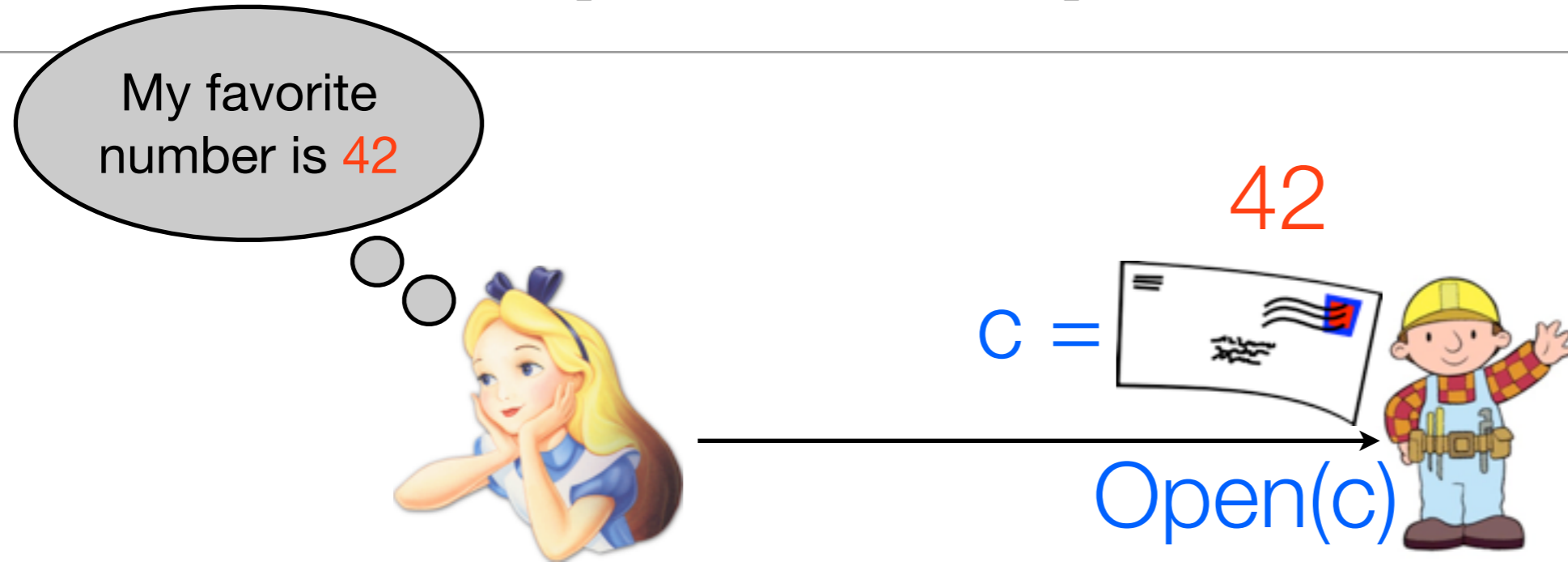
# Commitments [BCC88,P91]



There are two important properties of commitments:

- **Hiding**: Bob didn't know the value in  $c$  until Alice gave him  $\text{Open}(c)$

# Commitments [BCC88,P91]

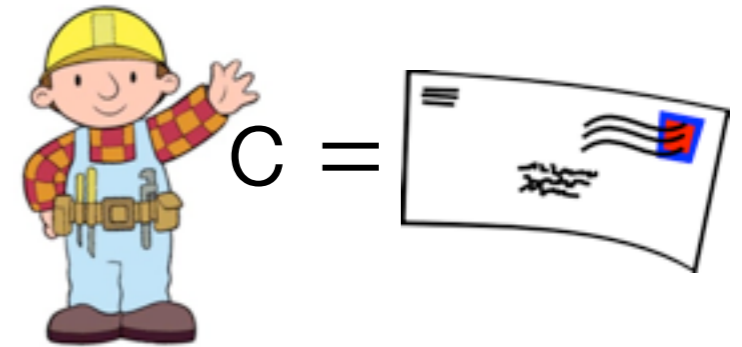


There are two important properties of commitments:

- **Hiding**: Bob didn't know the value in  $c$  until Alice gave him  $\text{Open}(c)$
- **Binding**: Alice couldn't change the value in  $c$  after giving Bob the envelope

# Zero-knowledge proofs [GMR89,BdSMP91]

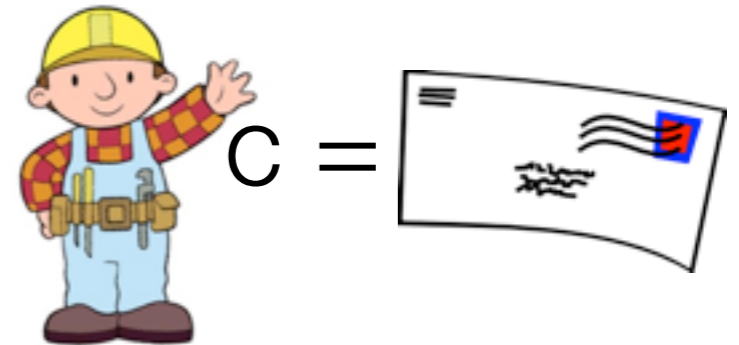
---



# Zero-knowledge proofs [GMR89,BdSMP91]

---

The value in  $c$  is between 0 and 100



# Zero-knowledge proofs [GMR89,BdSMP91]

---

The value in  $c$  is between 0 and 100

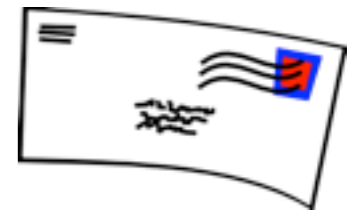


$\pi$



$c$

=



# Zero-knowledge proofs [GMR89,BdSMP91]

---





# Zero-knowledge proofs [GMR89,BdSMP91]

---



# Zero-knowledge proofs [GMR89,BdSMP91]

---



There are two important properties of zero-knowledge proofs:

# Zero-knowledge proofs [GMR89, BdSMP91]

---



There are two important properties of zero-knowledge proofs:

- **Soundness**: Alice can't convince Bob of something that isn't true

# Zero-knowledge proofs [GMR89,BdSMP91]

---



There are two important properties of zero-knowledge proofs:

- **Soundness**: Alice can't convince Bob of something that isn't true
- **Zero knowledge**: Bob doesn't learn anything about Alice's exact number

# Zero-knowledge proofs [GMR89,BdSMP91]

---



There are two important properties of zero-knowledge proofs:

- **Soundness**: Alice can't convince Bob of something that isn't true
- **Zero knowledge**: Bob doesn't learn anything about Alice's exact number

Zero-knowledge proofs are much more general than this, but this **range proof** is the only type we will need

# Blind identity-based encryption (IBE)

---



# Blind identity-based encryption (IBE)

---

Regular [S84,BF01,C01]:



# Blind identity-based encryption (IBE)

---

Regular [S84,BF01,C01]:



$$c = \text{Enc}(\text{"Bob"}, m)$$

→



# Blind identity-based encryption (IBE)

---

Regular [S84,BF01,C01]:



$$c = \text{Enc}(\text{"Bob"}, m)$$

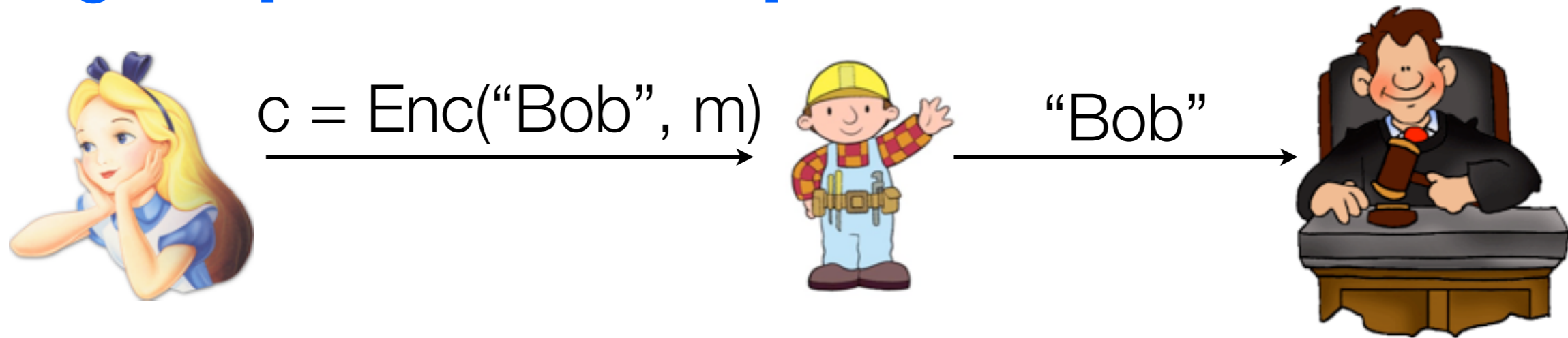
→



# Blind identity-based encryption (IBE)

---

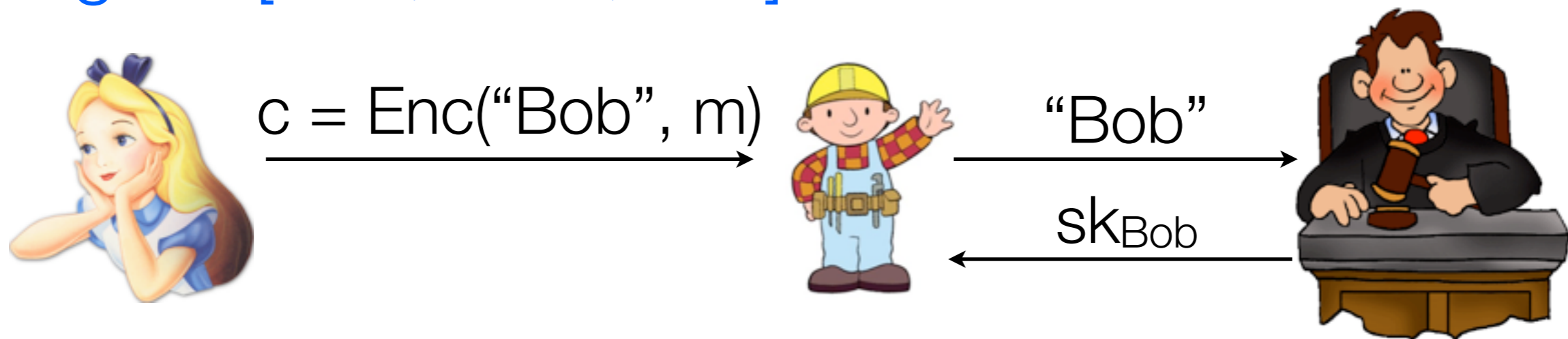
Regular [S84,BF01,C01]:



# Blind identity-based encryption (IBE)

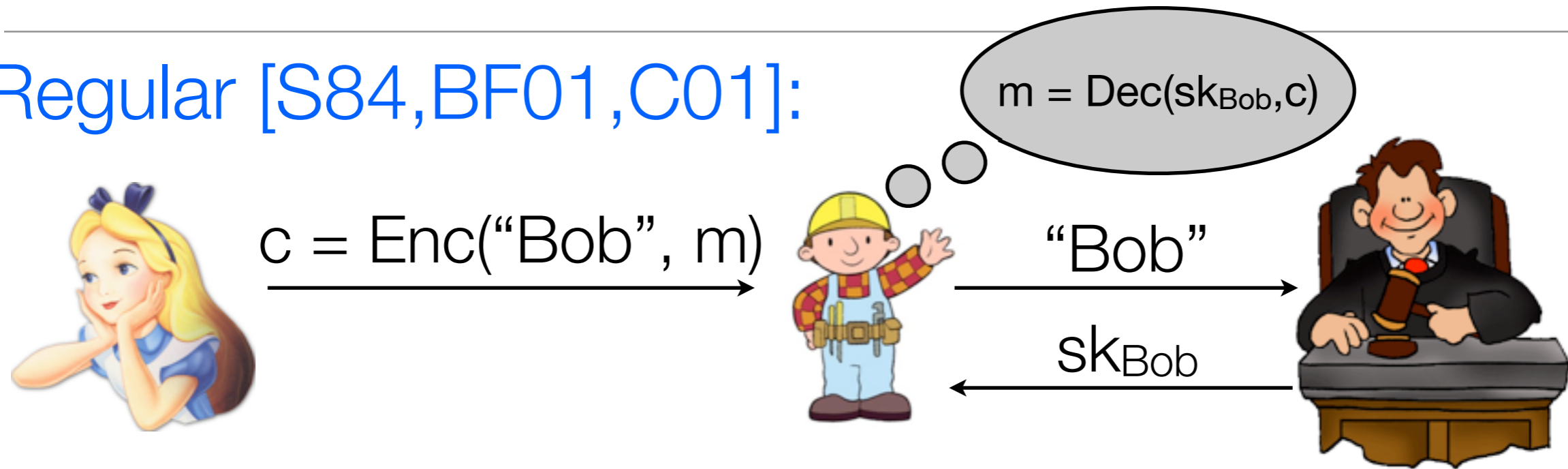
---

Regular [S84,BF01,C01]:



# Blind identity-based encryption (IBE)

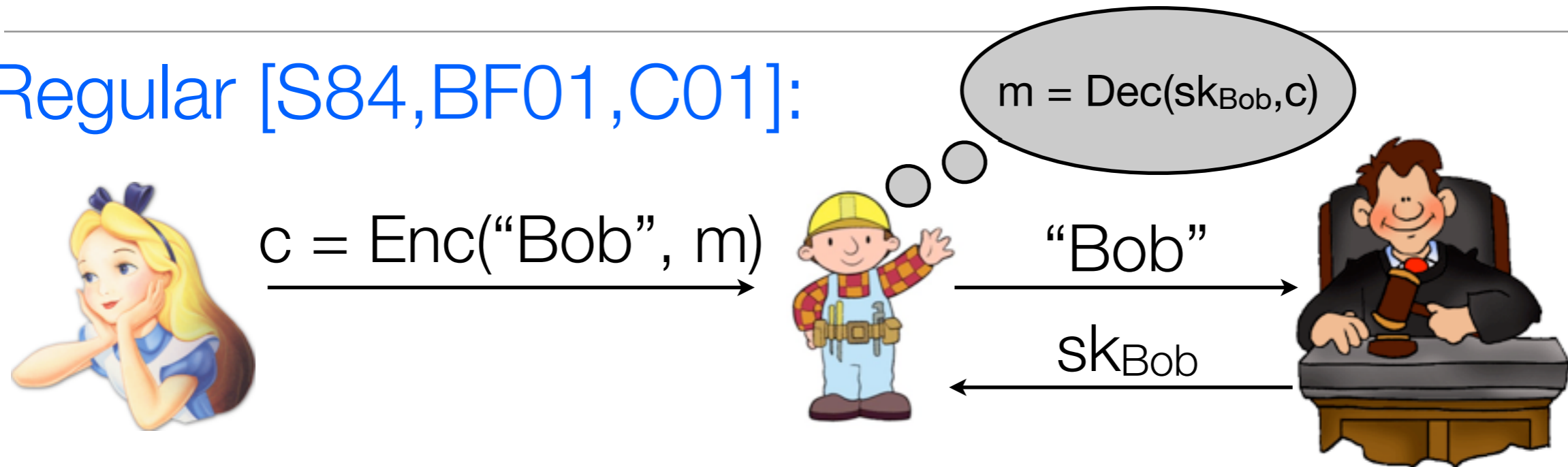
Regular [S84,BF01,C01]:





# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:

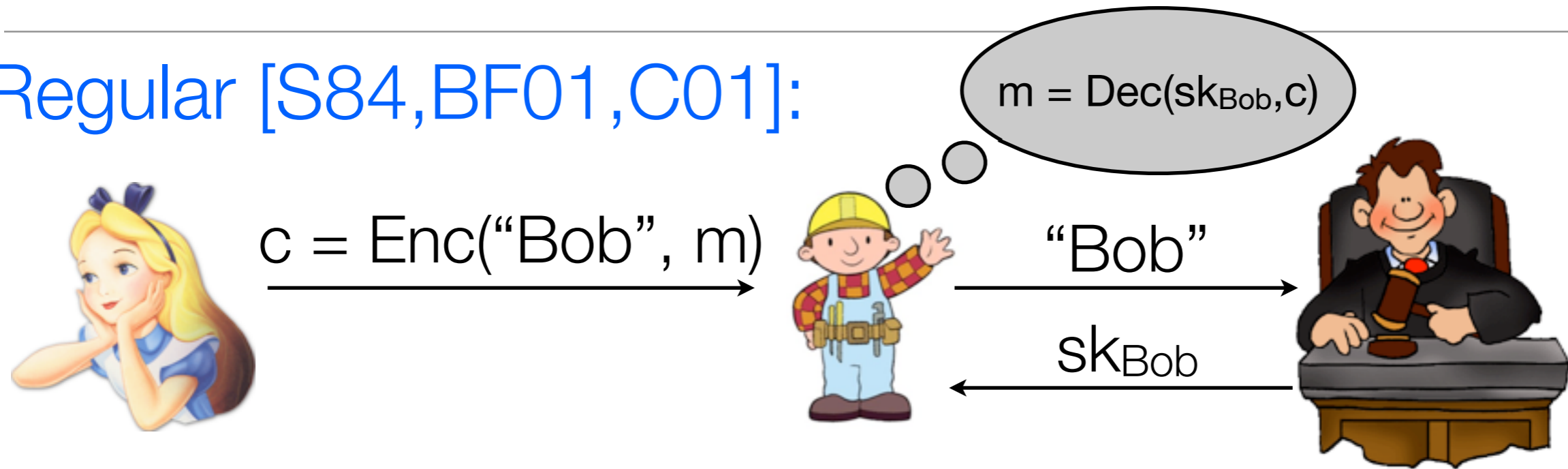


Blind [GH07]:

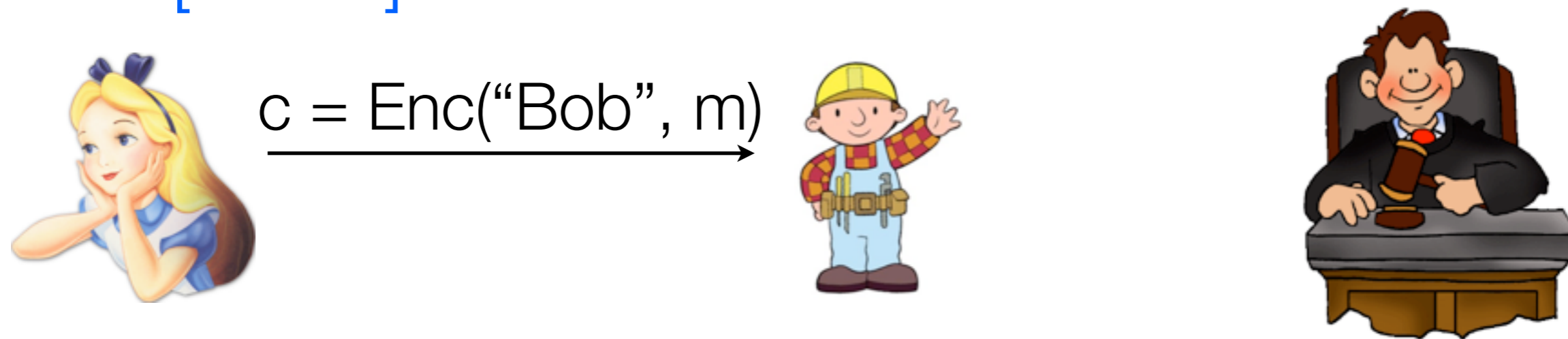


# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:

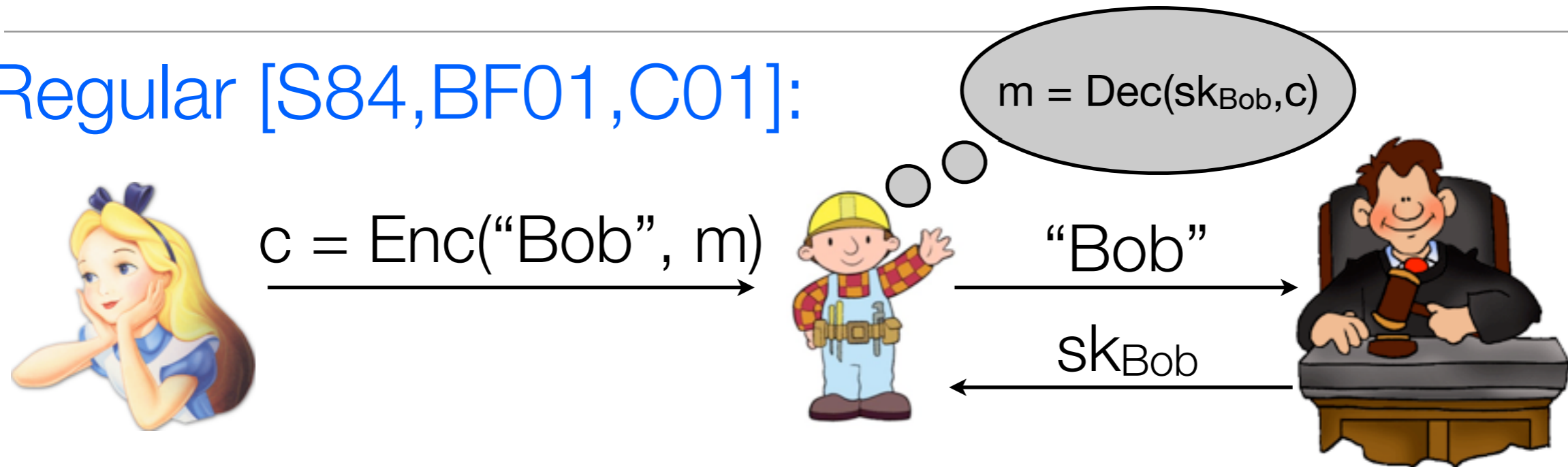


Blind [GH07]:

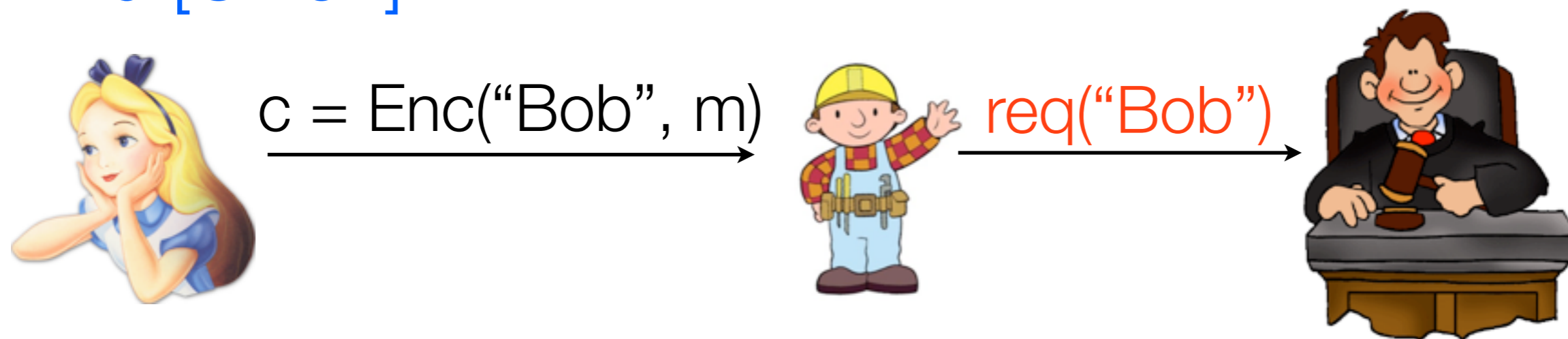


# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:

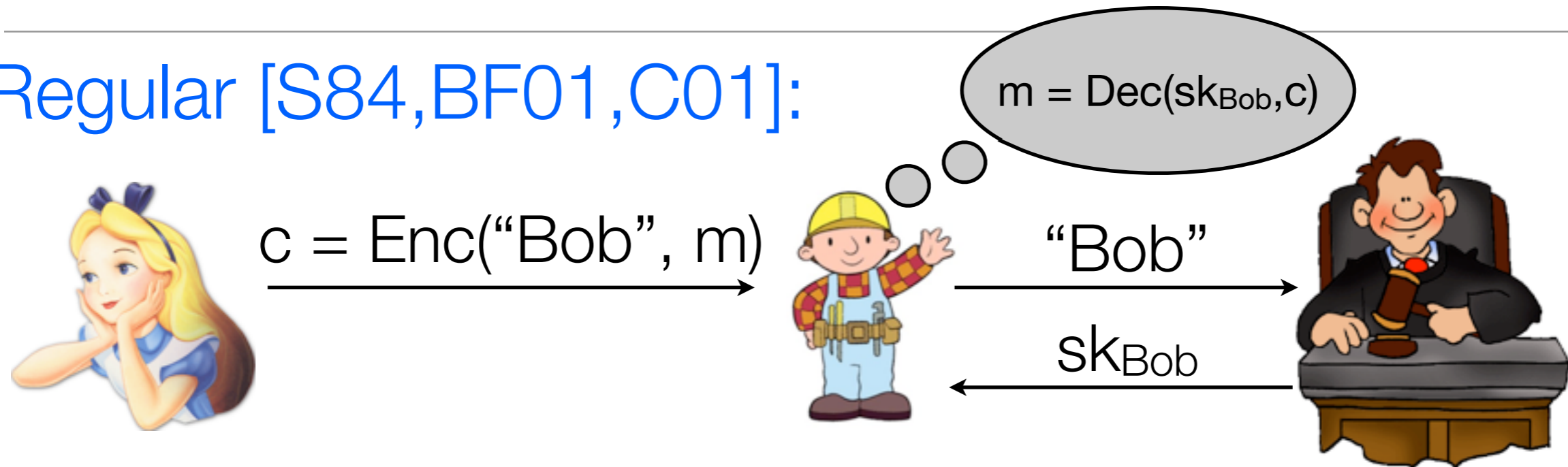


Blind [GH07]:

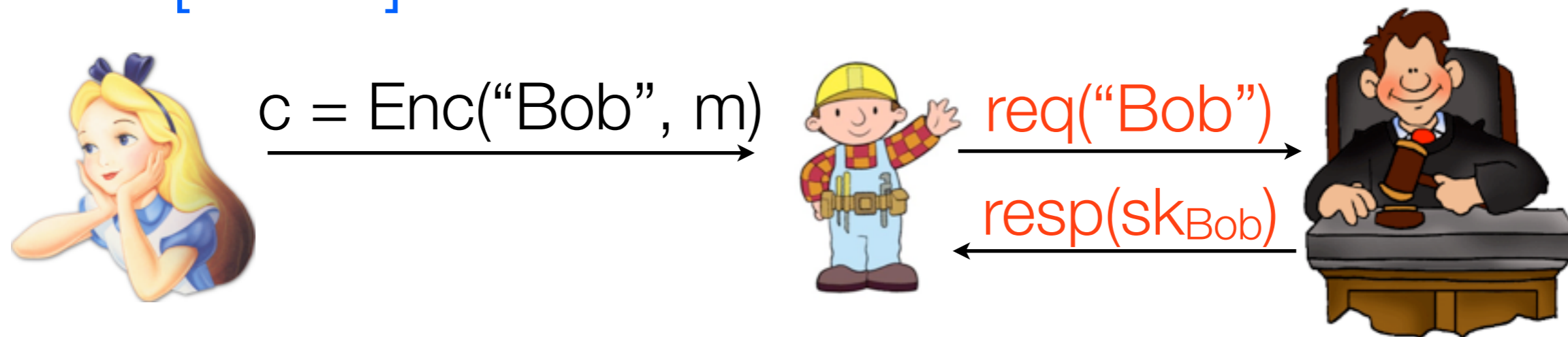


# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:



Blind [GH07]:



# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:



$$c = \text{Enc}(\text{"Bob"}, m)$$



"Bob"

$sk_{\text{Bob}}$



$$m = \text{Dec}(sk_{\text{Bob}}, c)$$

Blind [GH07]:



$$c = \text{Enc}(\text{"Bob"}, m)$$



$\text{req}(\text{"Bob"})$

$\text{resp}(sk_{\text{Bob}})$



1. Extract  $sk_{\text{Bob}}$  from resp
2.  $m = \text{Dec}(sk_{\text{Bob}}, c)$



# Blind identity-based encryption (IBE)

Regular [S84,BF01,C01]:



$$c = \text{Enc}(\text{"Bob"}, m)$$



"Bob"

$sk_{\text{Bob}}$



$$m = \text{Dec}(sk_{\text{Bob}}, c)$$

Blind [GH07]:



$$c = \text{Enc}(\text{"Bob"}, m)$$



$\text{req}(\text{"Bob"})$

$\text{resp}(sk_{\text{Bob}})$



1. Extract  $sk_{\text{Bob}}$  from resp
2.  $m = \text{Dec}(sk_{\text{Bob}}, c)$

So the authority **doesn't** learn which key is being extracted

# Outline

---

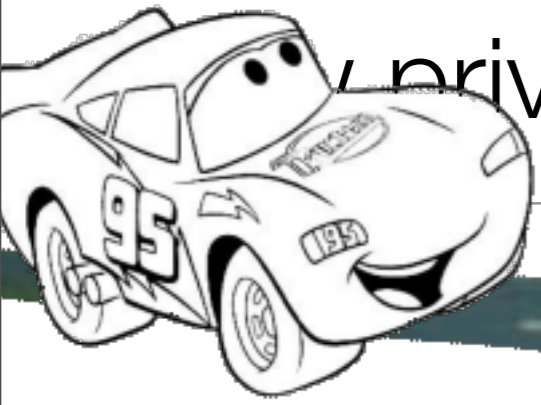
Cryptographic background

## Milo

A generic toll collection system  
A look back at (adapted) PrETP  
A new Audit protocol

Evaluation

Conclusions

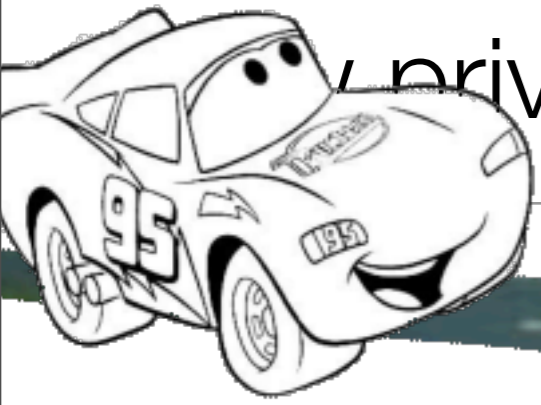


privacy-preserving toll pricing works

---







privacy-preserving toll pricing works

---



segments



privacy-preserving toll pricing works

A

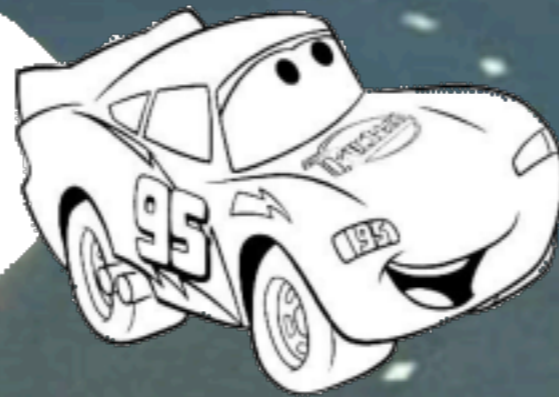


segments

# How privacy-preserving toll pricing works

---

A



segments

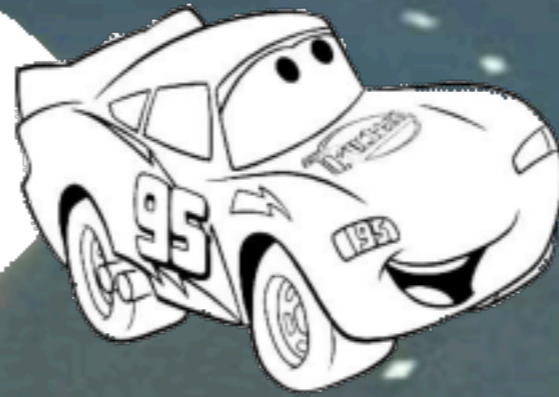


# How privacy-preserving toll pricing works

---

A

B



segments

# How privacy-preserving toll pricing works

---

A

B



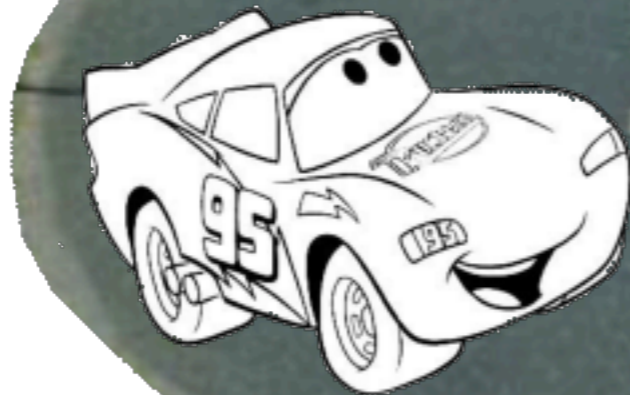
segments  
(A-B, 13:01-13:02)

# How privacy-preserving toll pricing works

---

A

B



segments  
(A-B, 13:01-13:02)



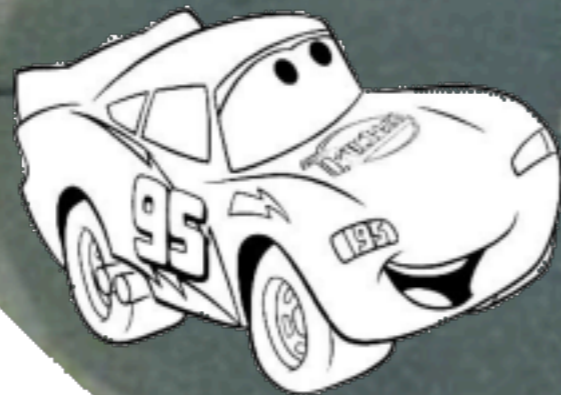
# How privacy-preserving toll pricing works

---

A

B

C



segments  
(A-B, 13:01-13:02)

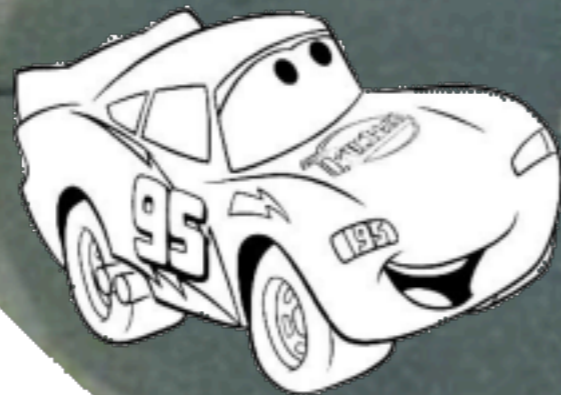
# How privacy-preserving toll pricing works

---

A

B

C



segments

(A-B, 13:01-13:02)

(B-C, 13:02-13:03)



# How privacy-preserving toll pricing works

---

A

B

C

segments

(A-B, 13:01-13:02)

(B-C, 13:02-13:03)



# How privacy-preserving toll pricing works

---

A

B

C

D

segments

(A-B, 13:01-13:02)

(B-C, 13:02-13:03)



# How privacy-preserving toll pricing works

---

A

B

C

D



segments

(A-B, 13:01-13:02)

(B-C, 13:02-13:03)

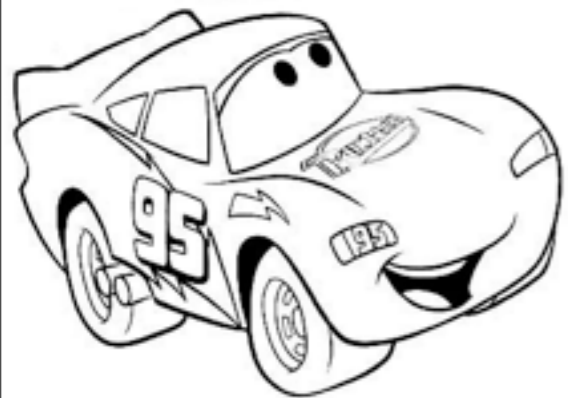
(C-D, 13:03-13:04)

# How privacy-preserving toll pricing works

---

# How privacy-preserving toll pricing works

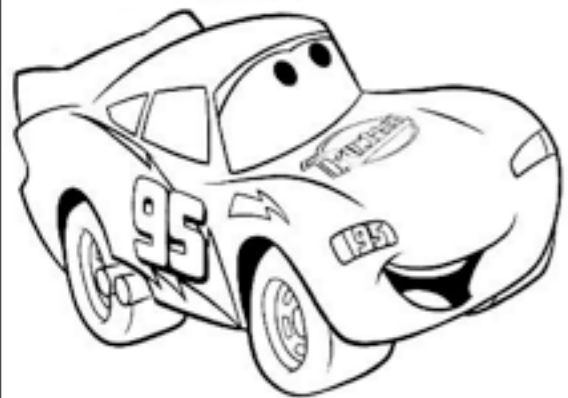
---



OBU

# How privacy-preserving toll pricing works

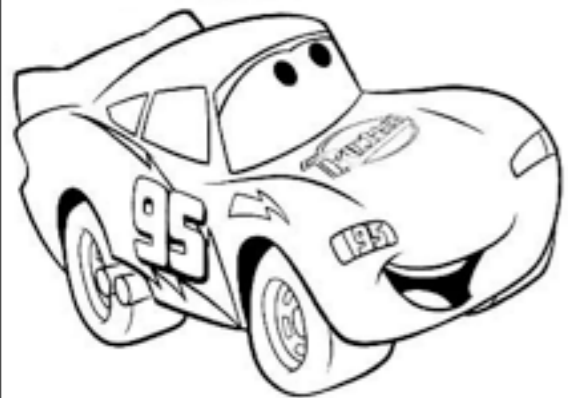
---



OBU  
segments

# How privacy-preserving toll pricing works

---



OBU  
segments

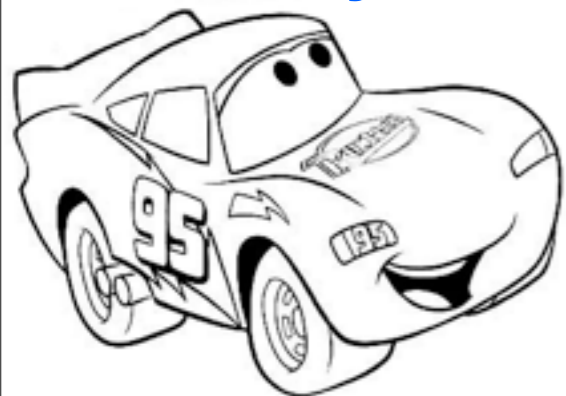


# How privacy-preserving toll pricing works

---



Payment

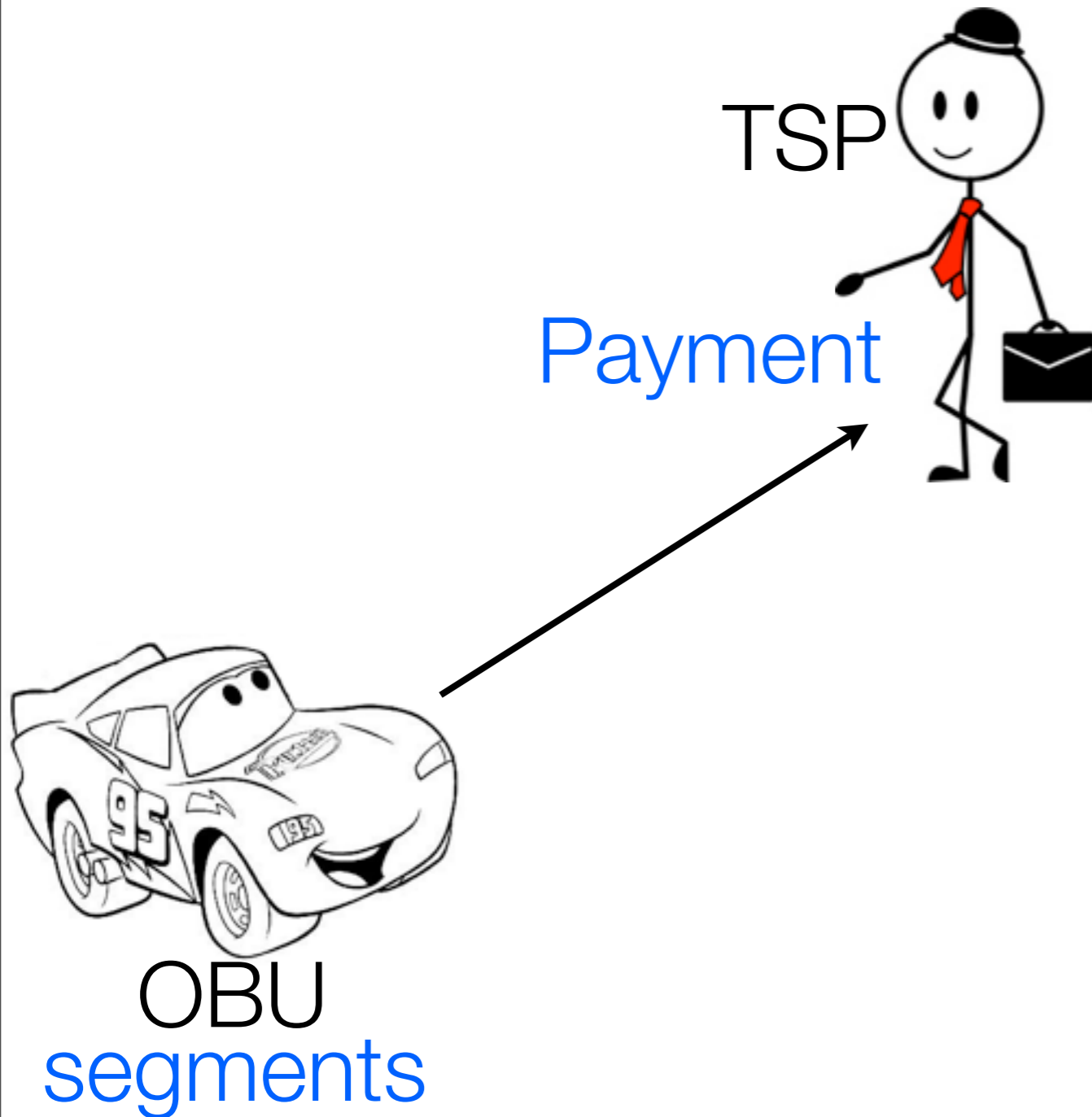


OBU  
segments



# How privacy-preserving toll pricing works

---

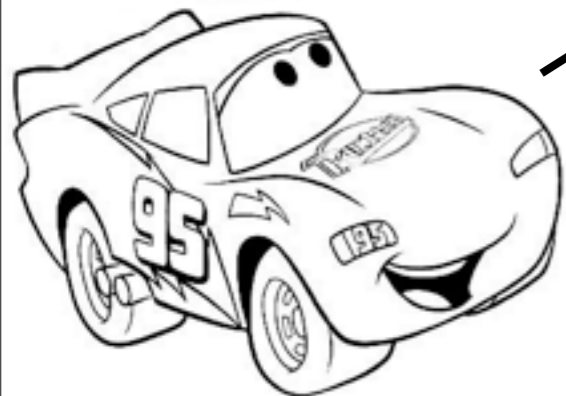


# How privacy-preserving toll

Check information and charge driver what they owe

TSP

Payment



OBU  
segments

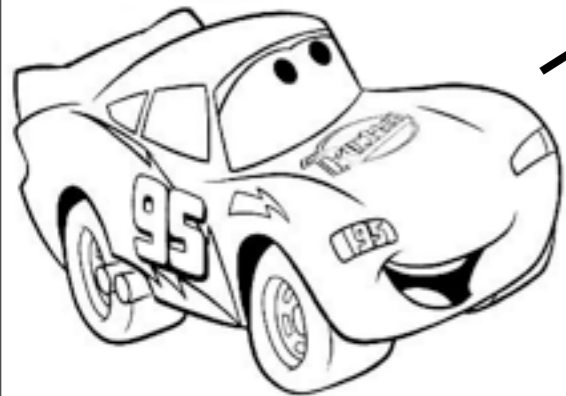
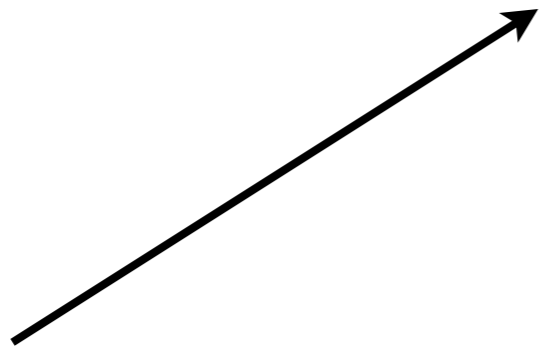
# How privacy-preserving toll

Check information and charge driver what they owe

TSP



Payment



OBU  
segments



TC

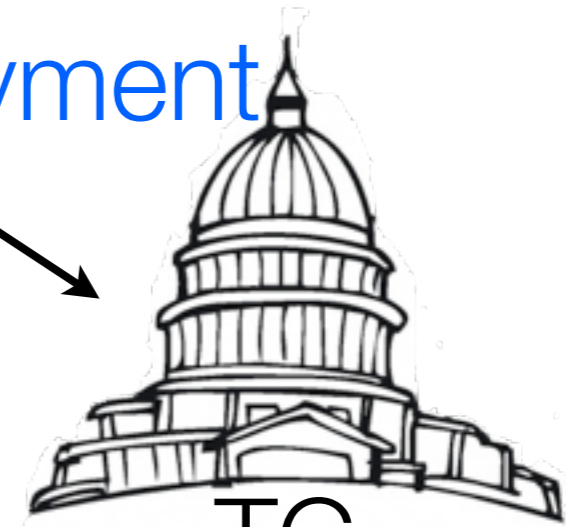
# How privacy-preserving toll

Check information and charge driver what they owe

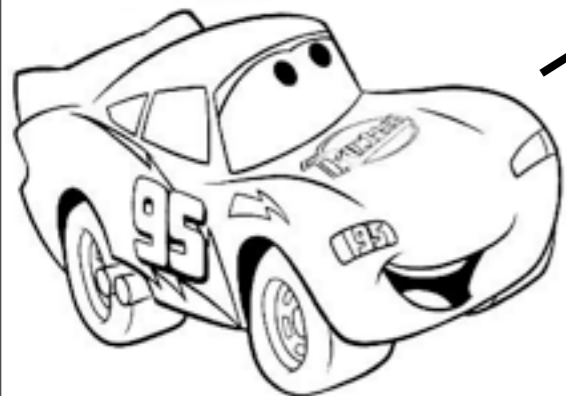
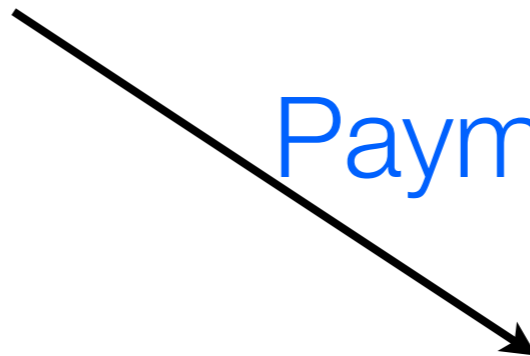
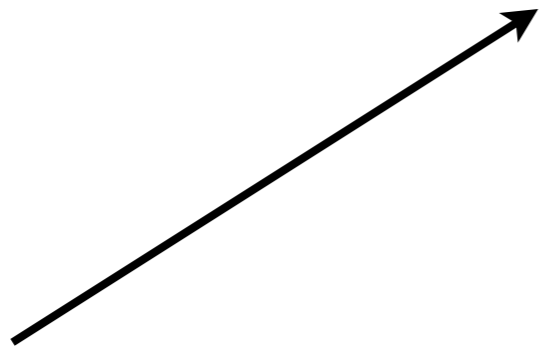
TSP



Payment



TC



OBU  
segments

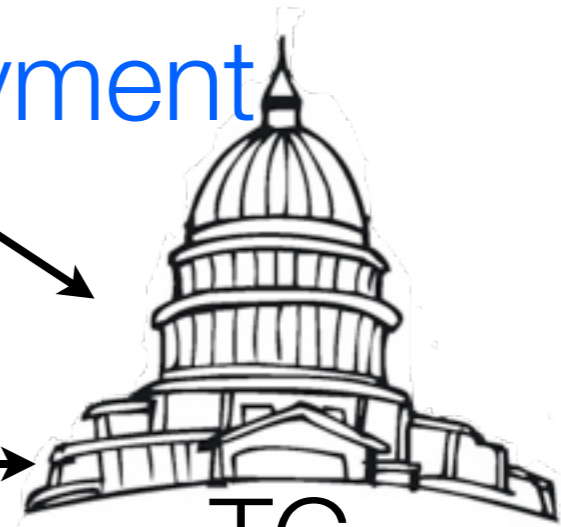
# How privacy-preserving toll

Check information and charge driver what they owe

TSP

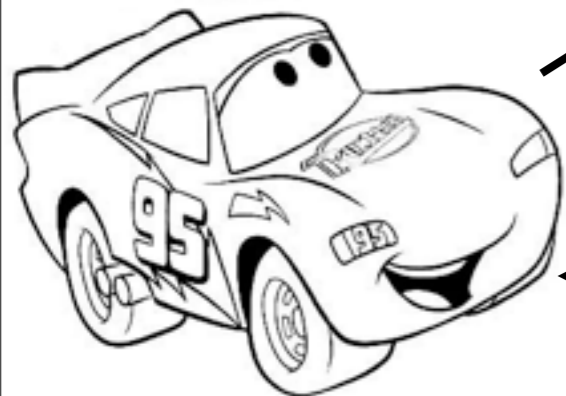


Payment



TC

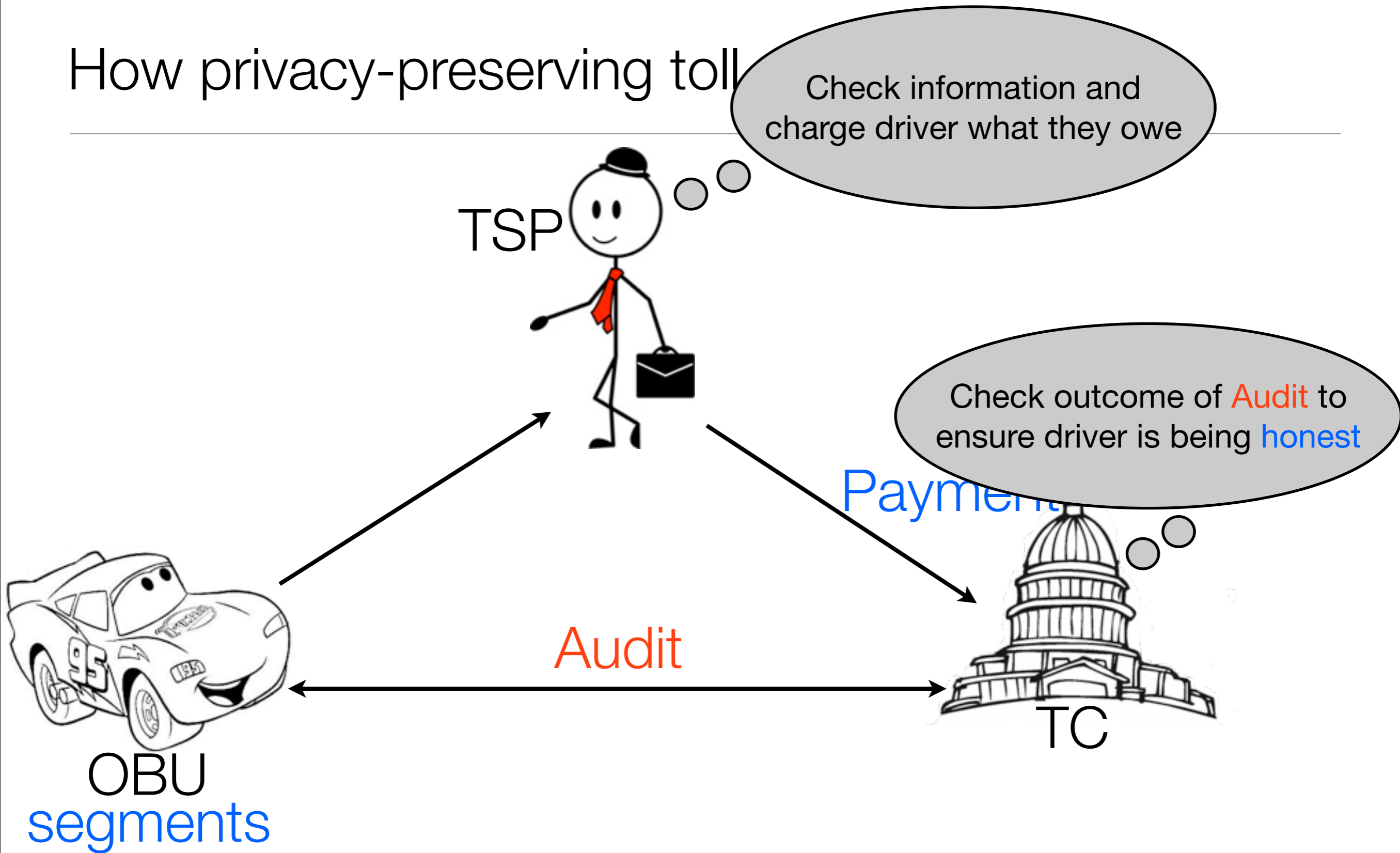
Audit



OBU

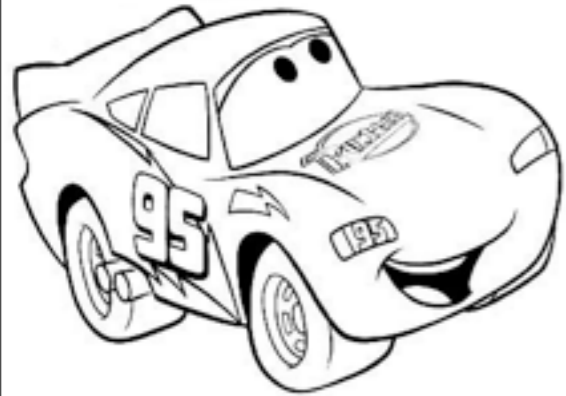
segments

# How privacy-preserving toll



# An adapted version of PrETP

---



# An adapted version of PrETP

---



$\{C_i, \pi_i\}_i$





# An adapted version of PrETP

---



Commitment  
to segment  
price  $p_i$

$\{C_i, \pi_i\}_i$



# An adapted version of PrETP

---



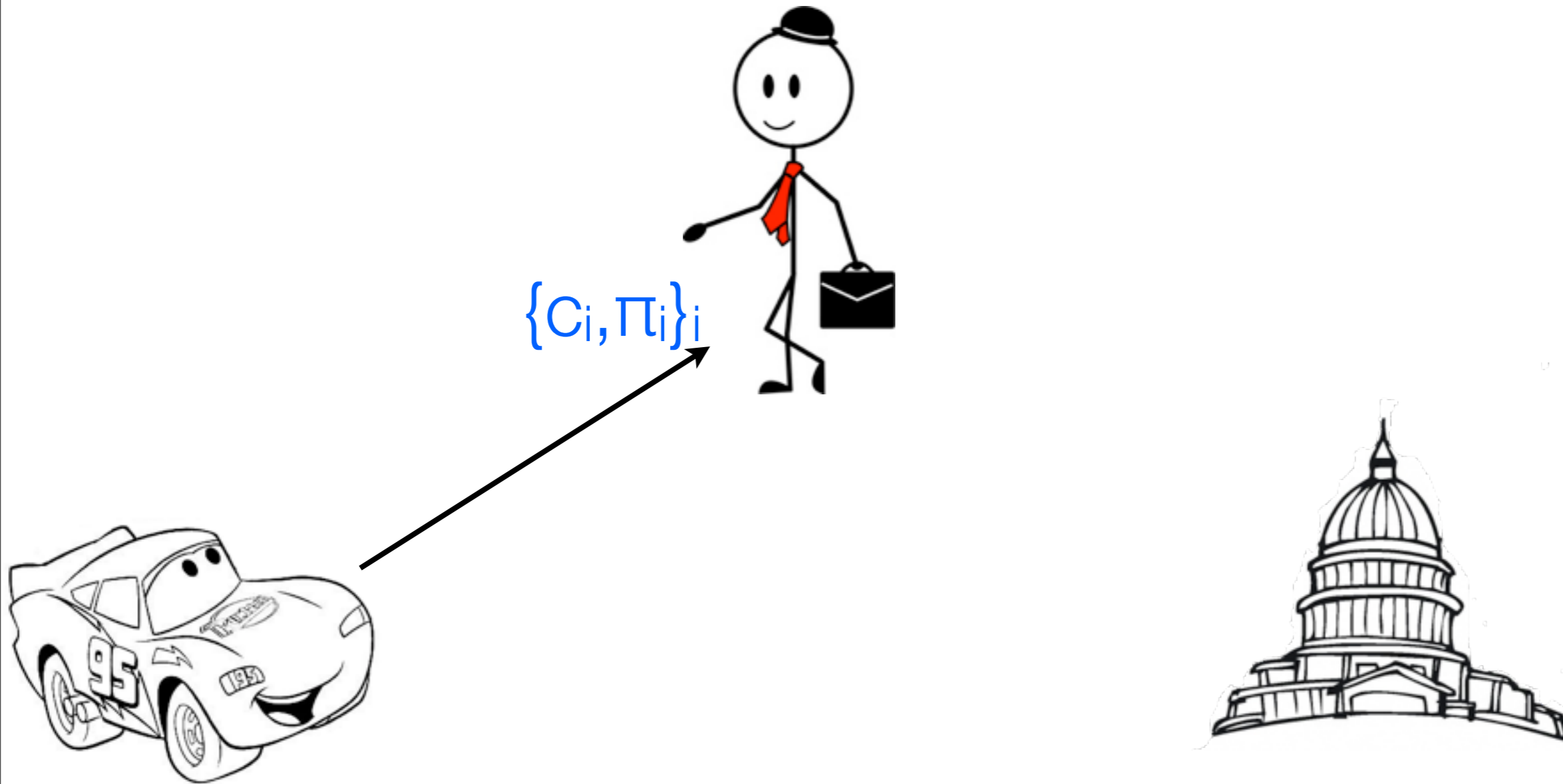
Commitment to segment price  $p_i$       NIZK that the value in  $c_i$  is in the proper range

$\{c_i, \pi_i\}_i$



# An adapted version of PrETP

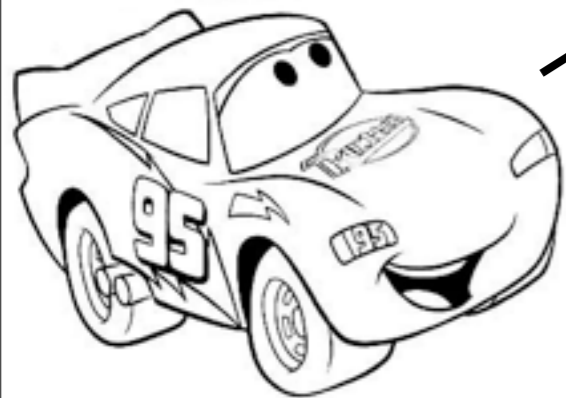
---



# An adapted version of PrETP

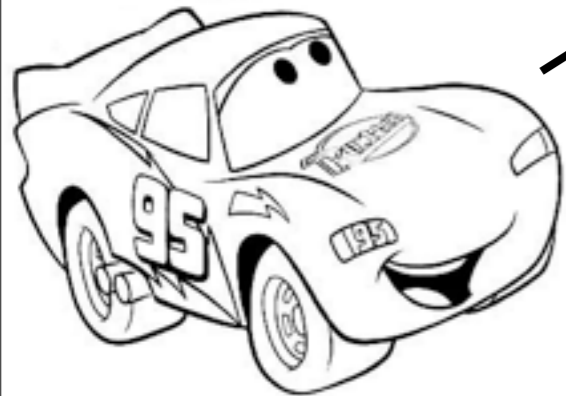
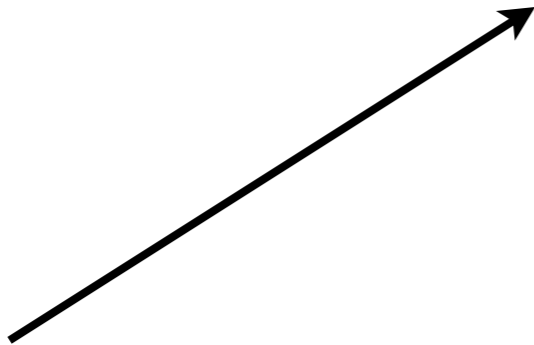
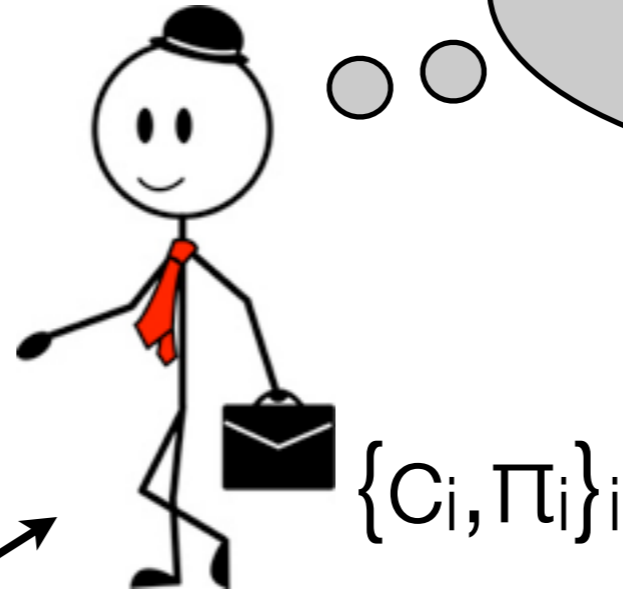
1. Verify each NIZK  $\pi_i$
2. Compute total price

$\{C_i, \pi_i\}_i$



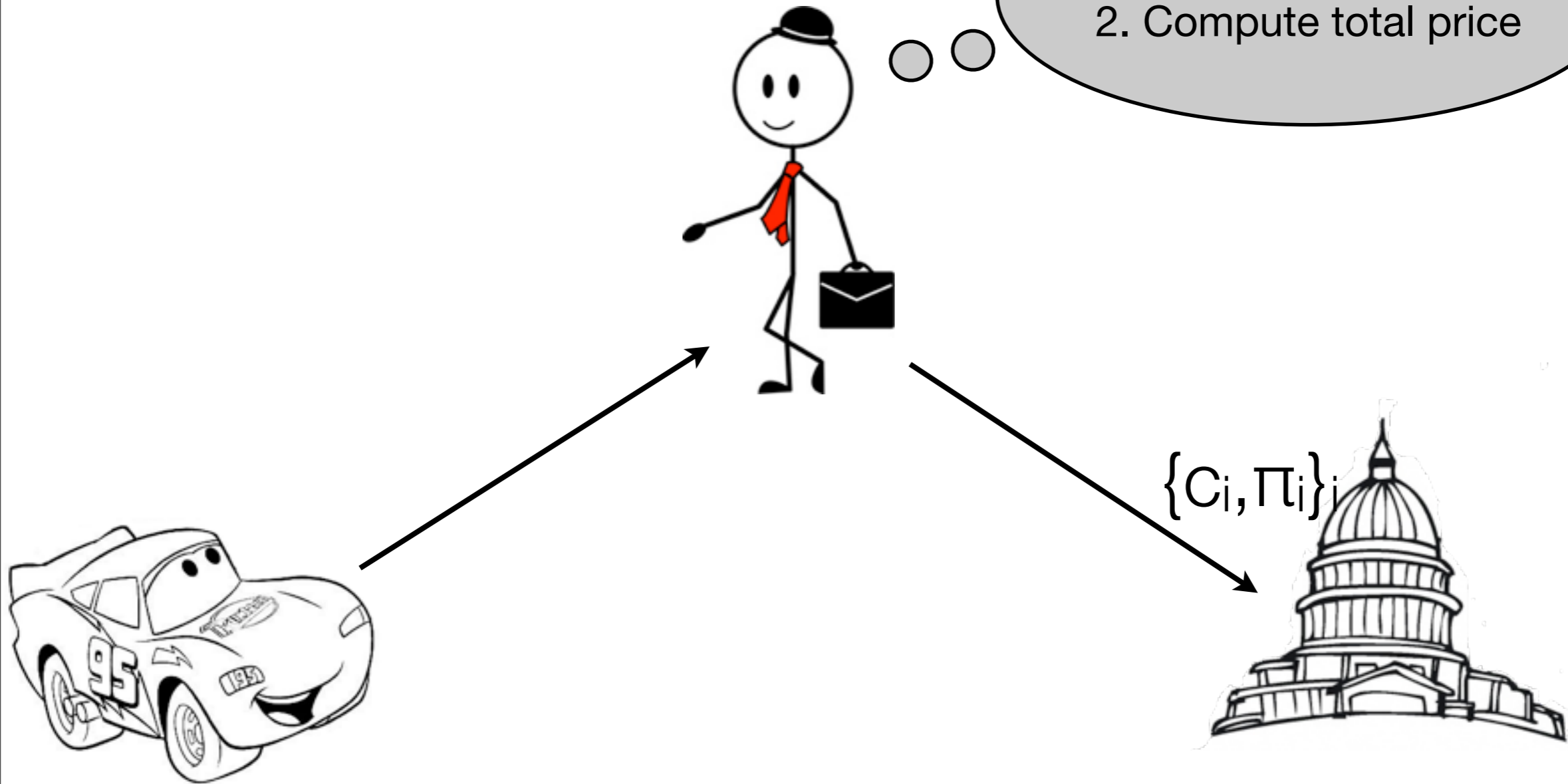
# An adapted version of PrETP

1. Verify each NIZK  $\pi_i$
2. Compute total price



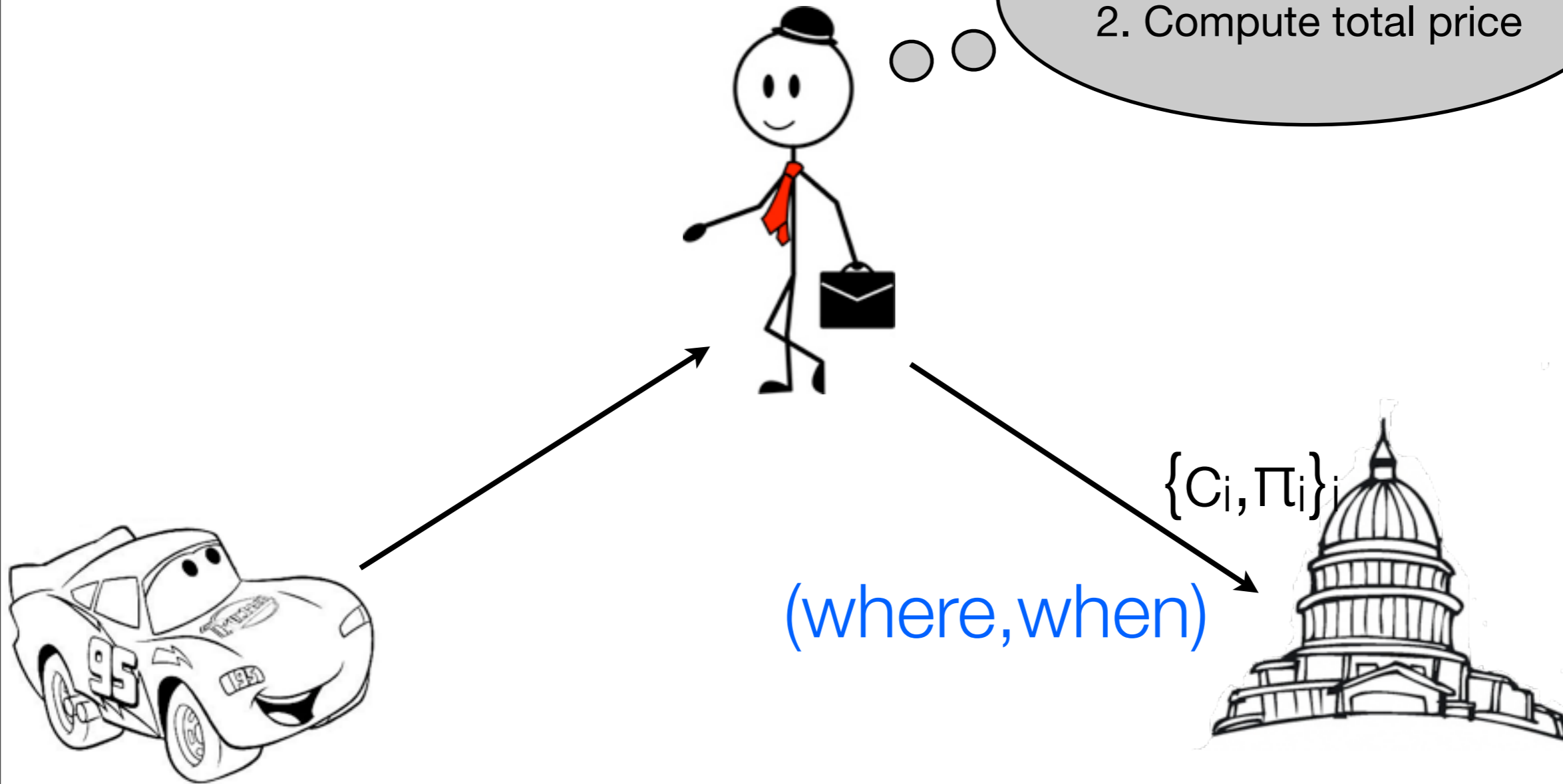
# An adapted version of PrETP

1. Verify each NIZK  $\pi_i$
2. Compute total price



# An adapted version of PrETP

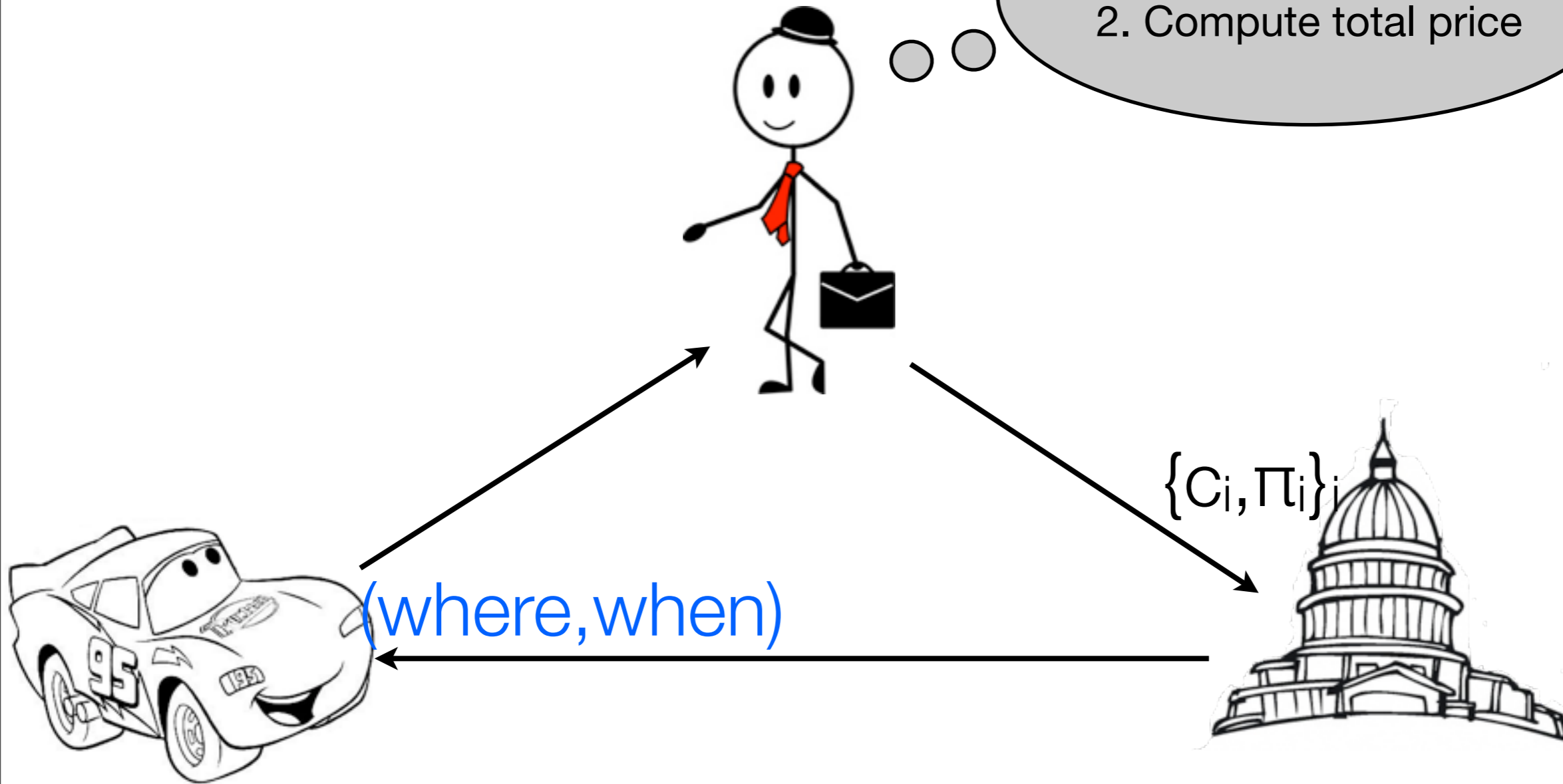
1. Verify each NIZK  $\pi_i$
2. Compute total price





# An adapted version of PrETP

1. Verify each NIZK  $\pi_i$
2. Compute total price

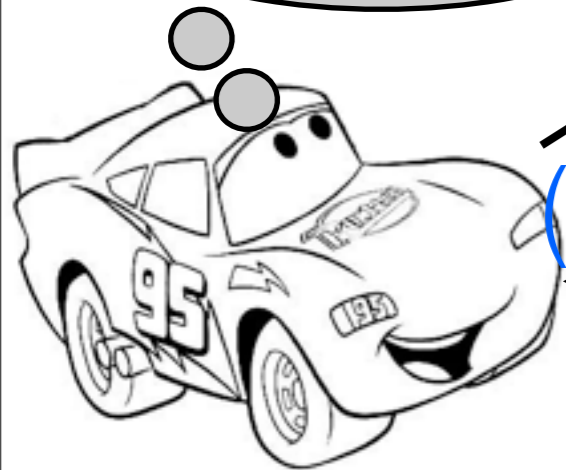


# An adapted version of PrETP

1. Verify each NIZK  $\pi_i$
2. Compute total price

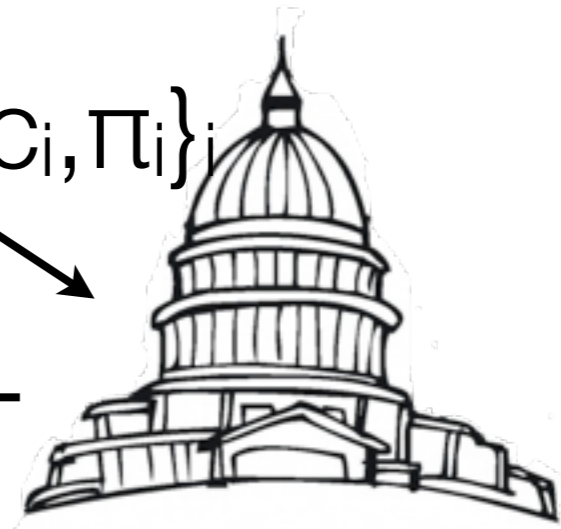


Find  
commitment  $c_j$  for  
(where, when)



(where, when)

$\{c_i, \pi_i\}_i$

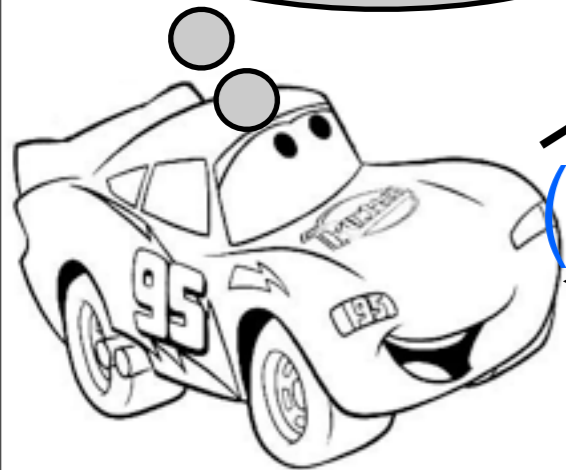


# An adapted version of PrETP

1. Verify each NIZK  $\pi_i$
2. Compute total price

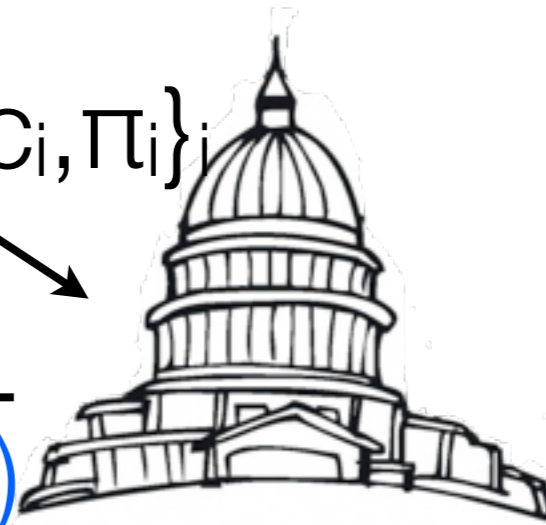


Find commitment  $c_j$  for (where, when)



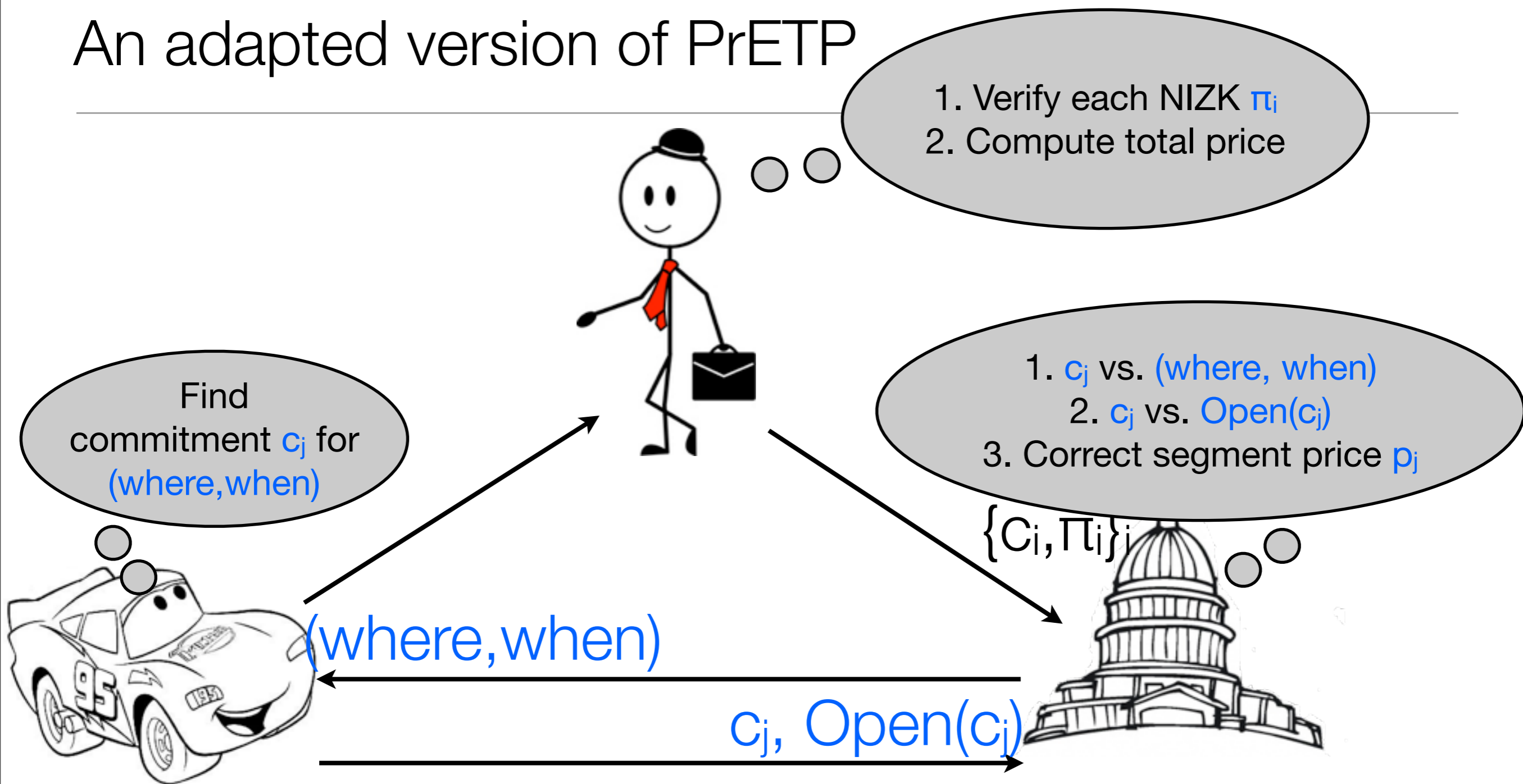
(where, when)

$\{c_i, \pi_i\}_i$

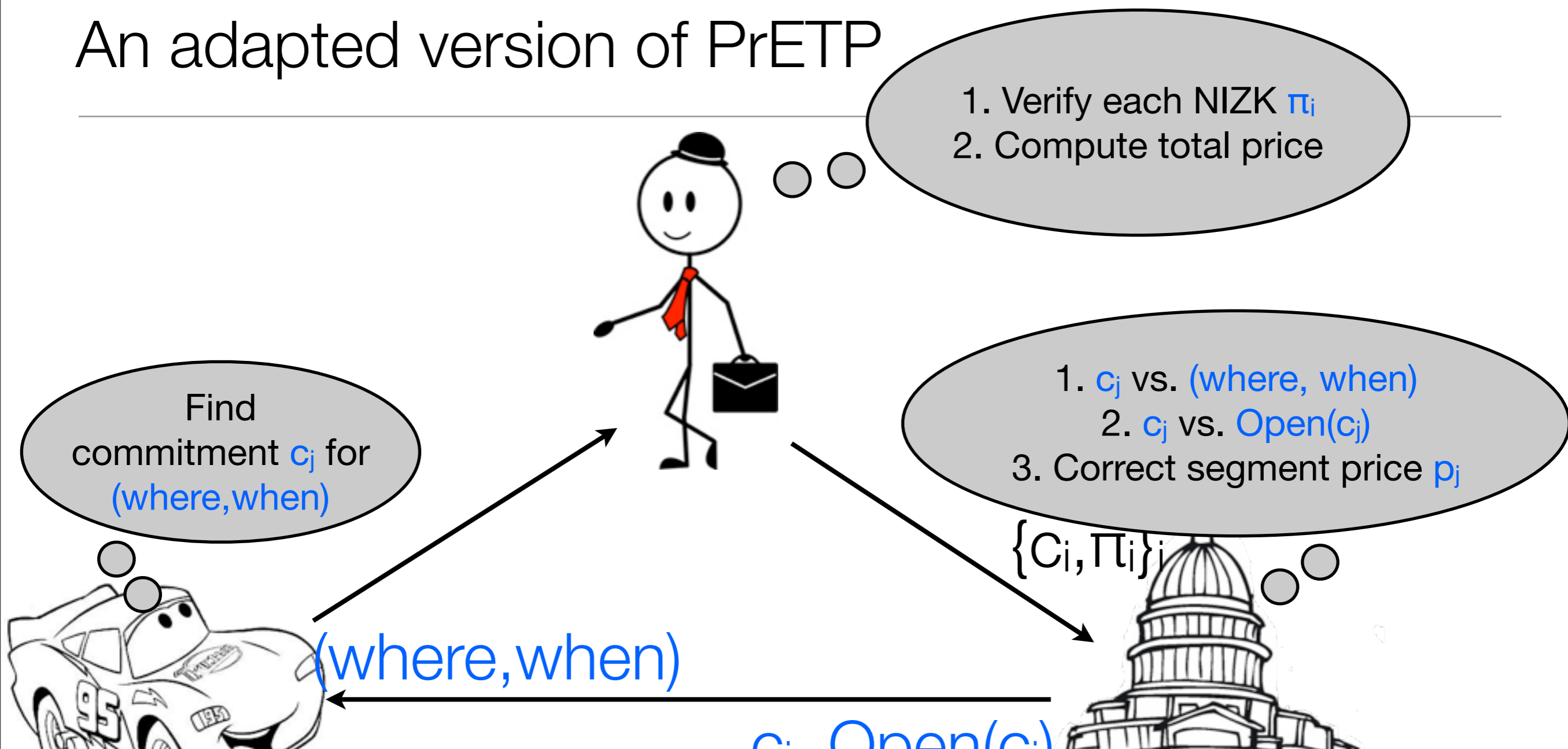


$c_j, \text{Open}(c_j)$

# An adapted version of PrETP



# An adapted version of PrETP



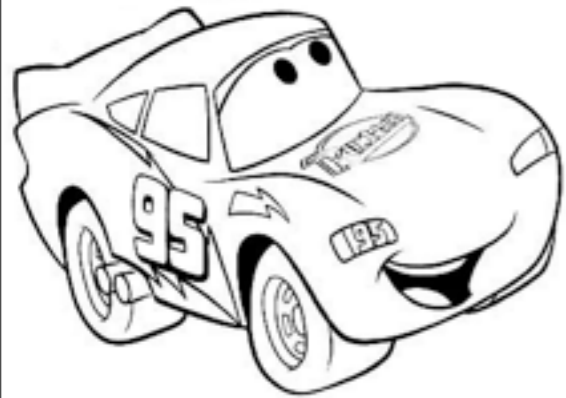
NIZK **zero knowledge** and commitment **hiding** guarantee driver privacy

NIZK **soundness** guarantees price  $p_i$  is in the right range (e.g., non-negative)

Commitment **binding** guarantees  $c_j$  is the right commitment for  $(where,when)$

# “PrETP with sugar on top”: our new Audit protocol

---

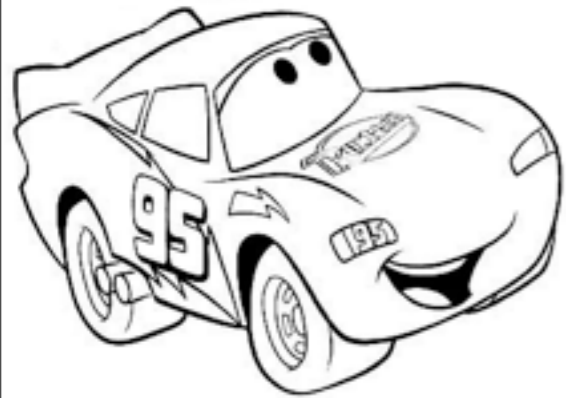


# “PrETP with sugar on top”: our new Audit protocol

---



$\{C_i, C_i, \Pi_i\}_i$



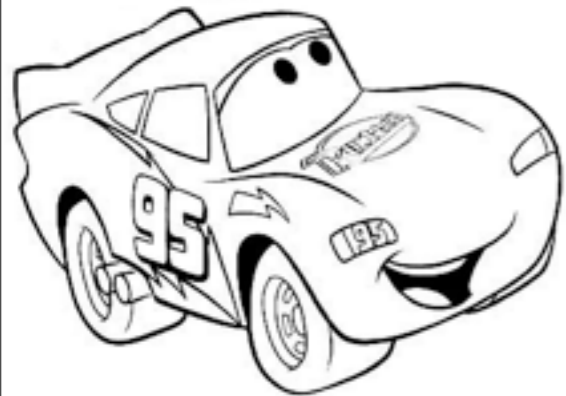


# “PrETP with sugar on top”: our new Audit protocol

---

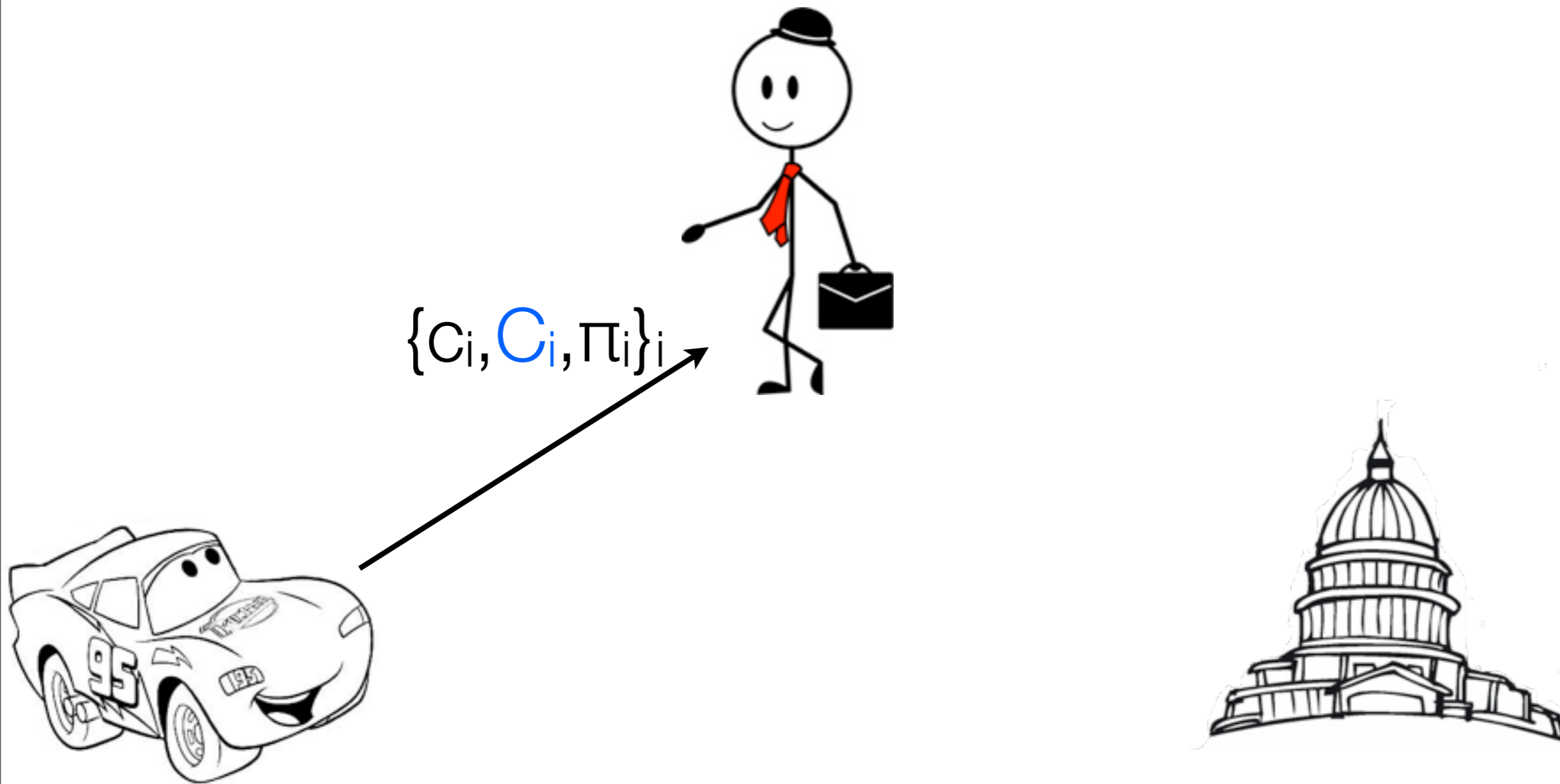
Blind IBE of the opening to  $c_i$ , using (where, when) as identity

$\{c_i, C_i, \pi_i\}_i$



# “PrETP with sugar on top”: our new Audit protocol

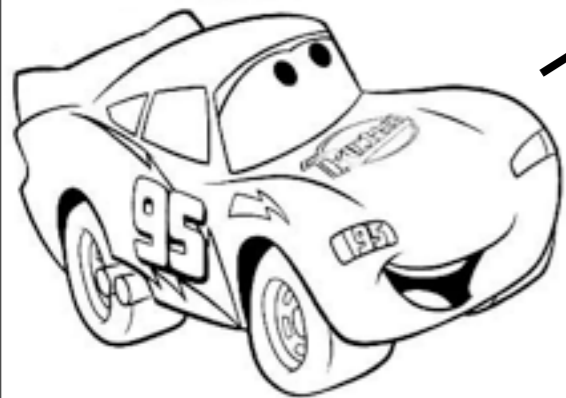
---



# “PrETP with sugar on top”: our new Audit protocol

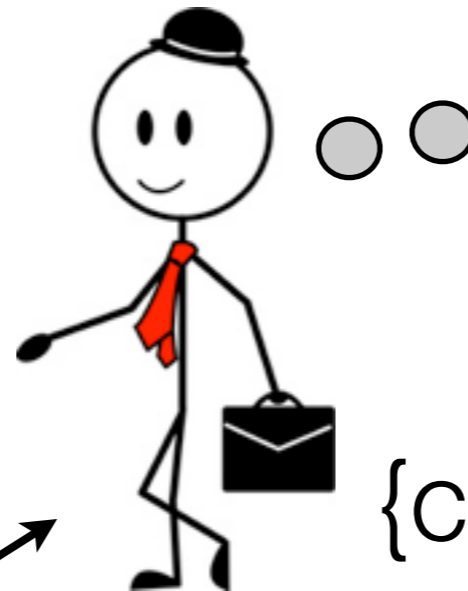
1. Verify each NIZK  $\pi_i$
2. Compute total price

$\{c_i, C_i, \pi_i\}_i$

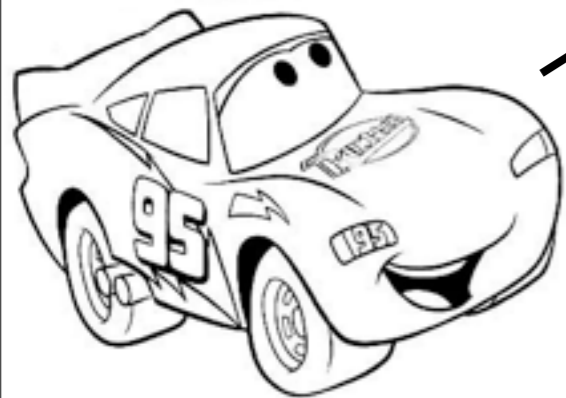


# “PrETP with sugar on top”: our new $\Delta$ -audit protocol

1. Verify each NIZK  $\pi_i$
2. Compute total price

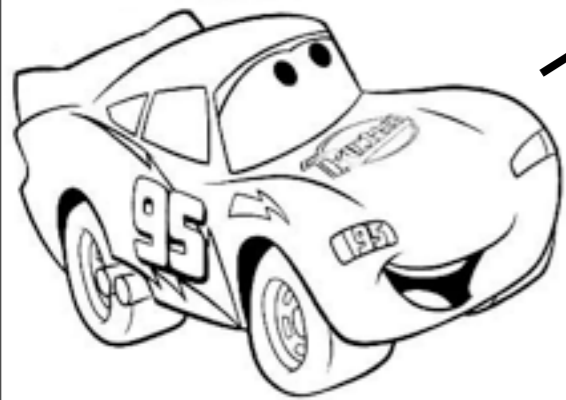


$\{c_i, C_i, \pi_i\}_i$

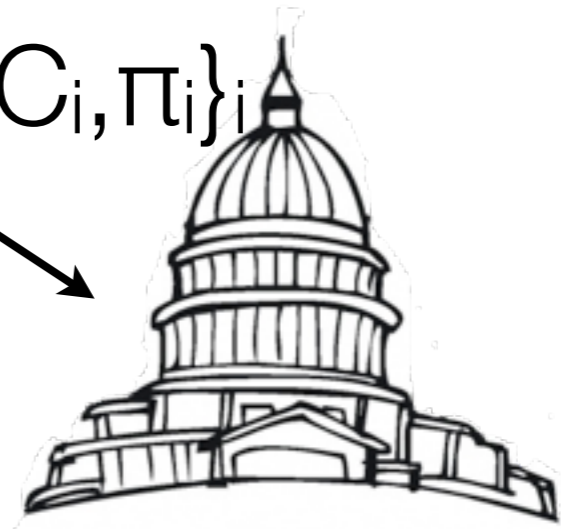


# “PrETP with sugar on top”: our new $\Delta$ -audit protocol

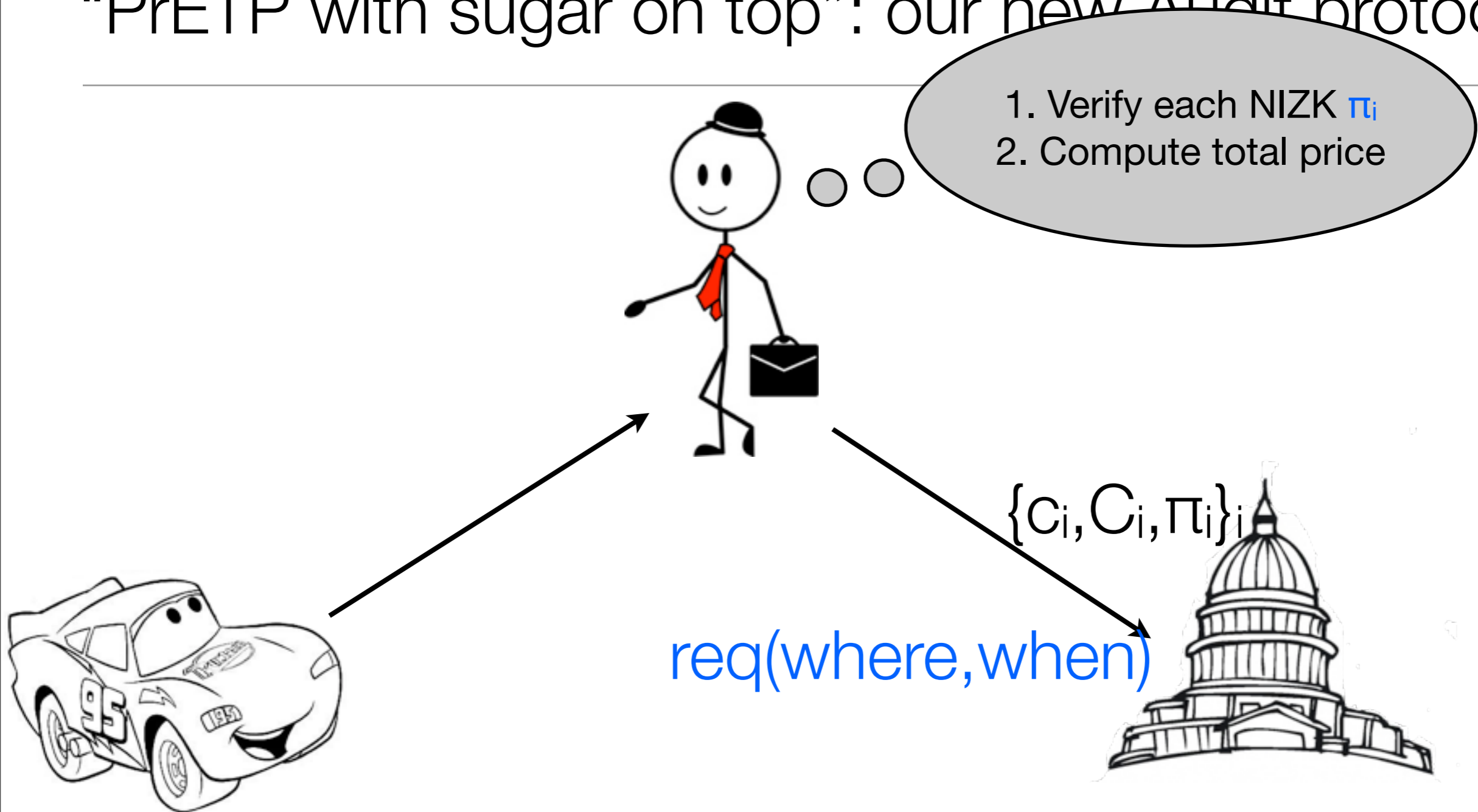
1. Verify each NIZK  $\pi_i$
2. Compute total price



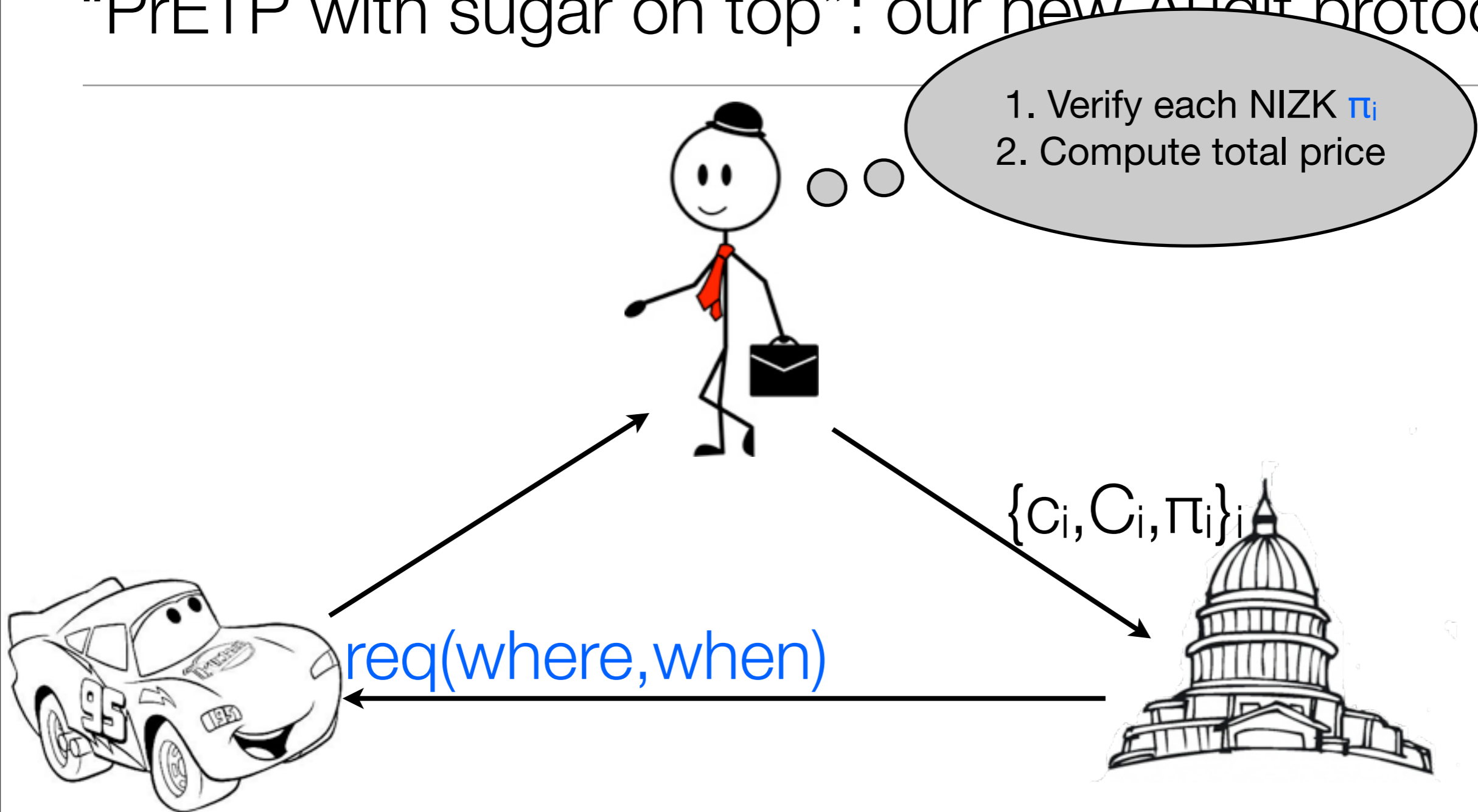
$\{c_i, C_i, \pi_i\}_i$



# “PrETP with sugar on top”: our new $\Delta$ -audit protocol

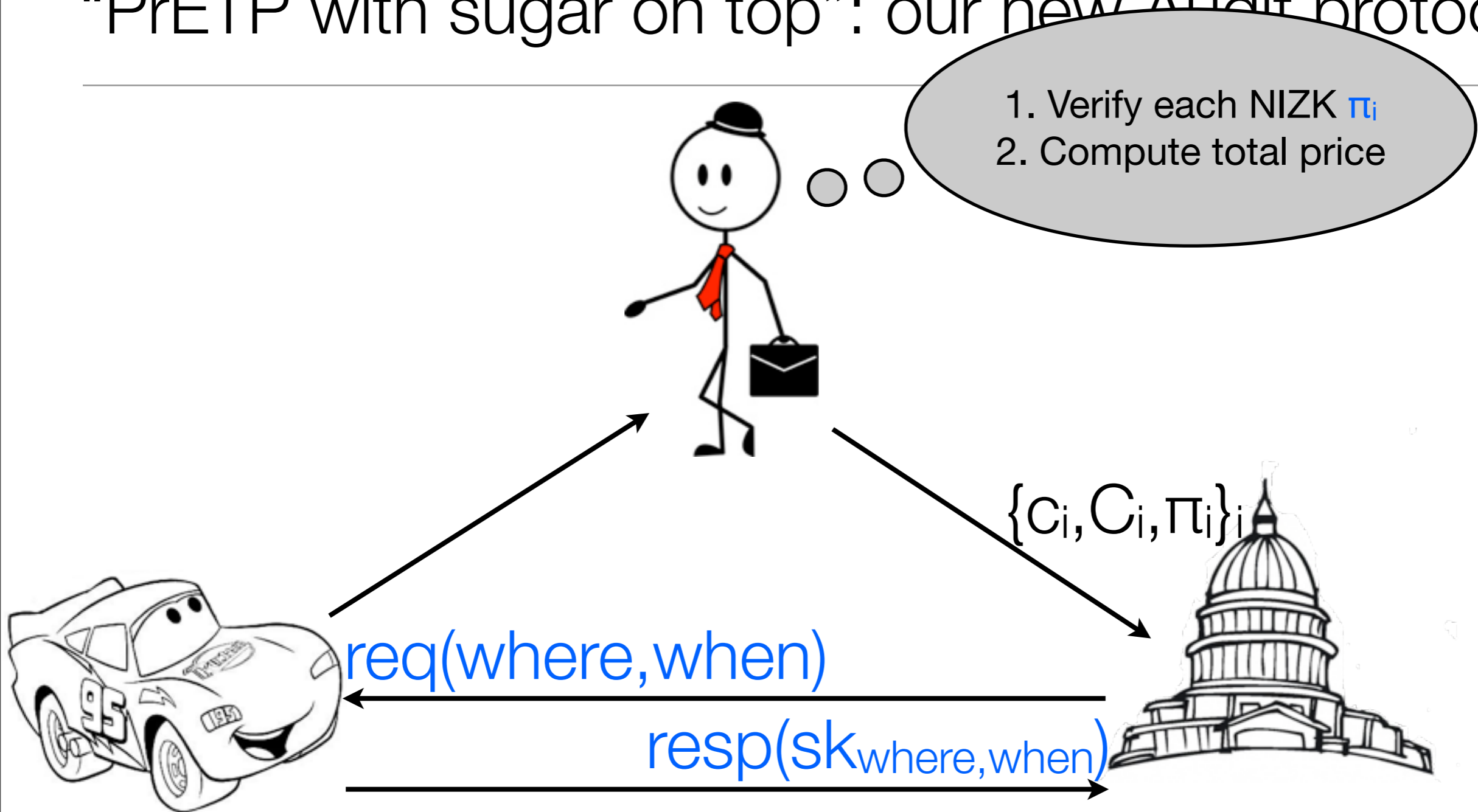


# “PrETP with sugar on top”: our new $\Delta$ -audit protocol

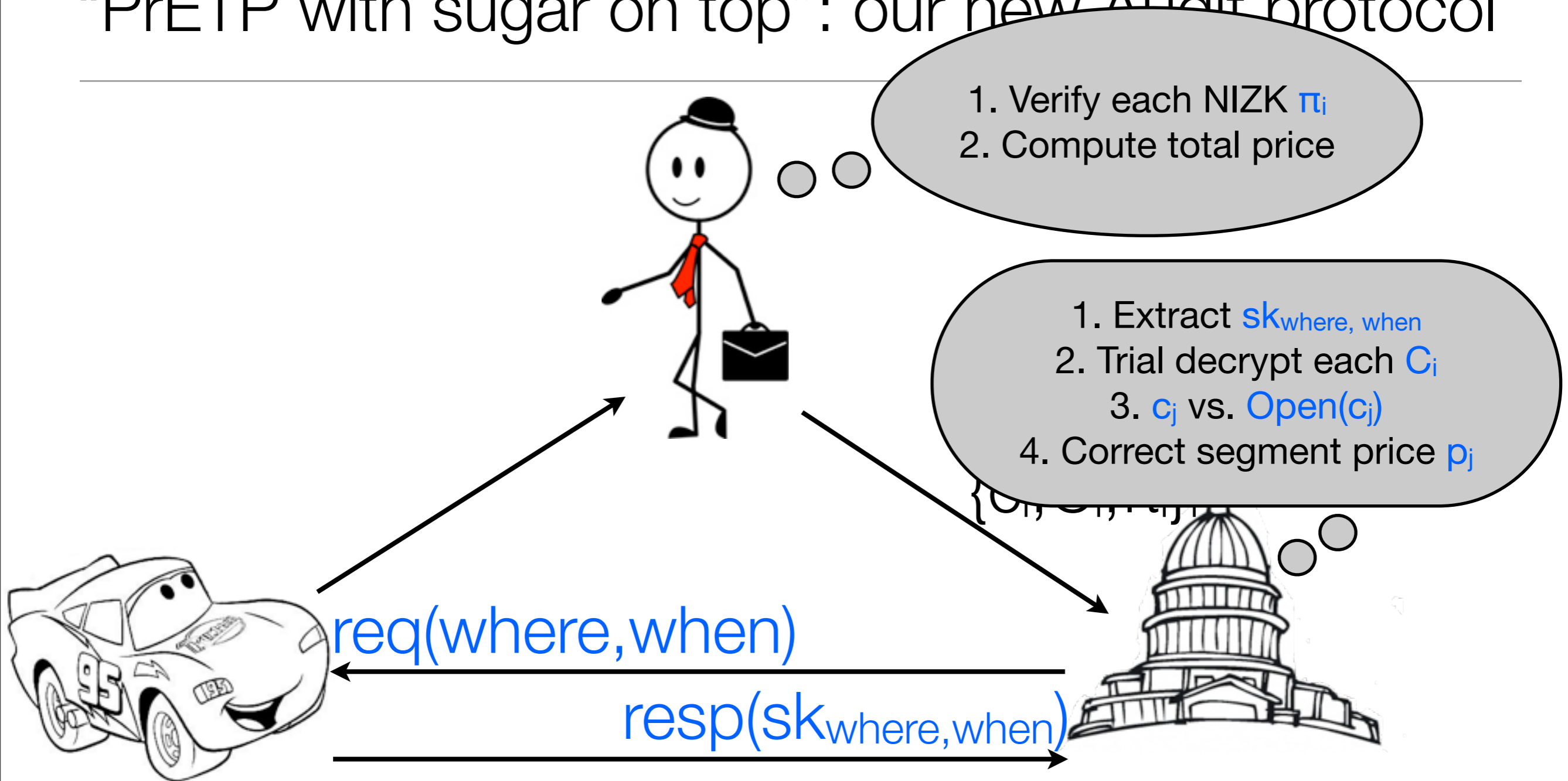




# “PrETP with sugar on top”: our new $\Delta$ -audit protocol



# “PrETP with sugar on top”: our new Audit protocol



# “PrETP with sugar on top”: our new Audit protocol



1. Verify each NIZK  $\pi_i$
2. Compute total price

1. Extract  $sk_{where, when}$
2. Trial decrypt each  $C_i$
3.  $c_j$  vs.  $Open(c_j)$
4. Correct segment price  $p_j$

$req(where, when)$

NIZK **zero knowledge** and commitment **hiding** guarantee driver privacy

NIZK **soundness** guarantees price  $p_i$  is in the right range (e.g., non-negative)

Commitment **binding** guarantees  $c_j$  is the right commitment for  $(where, when)$

# “PrETP with sugar on top”: our new Audit protocol



1. Verify each NIZK  $\pi_i$
2. Compute total price

1. Extract  $sk_{where, when}$
2. Trial decrypt each  $C_i$
3.  $c_j$  vs.  $Open(c_j)$
4. Correct segment price  $p_j$



$req(when, where)$



NIZK **zero knowledge** and commitment **hiding** guarantee driver privacy

NIZK **soundness** guarantees price  $p_i$  is in the right range (e.g., non-negative)

Commitment **binding** guarantees  $c_j$  is the right commitment for  $(when, where)$

IBE **blindness** guarantees that driver doesn't learn segment  $(when, where)$

# Outline

---

Cryptographic background

Milo

**Evaluation**

Implementation details  
Milo's performance

Conclusions

# Implementation

---

# Implementation

---

Used [MIRACL](#) [Scott] for blind IBE, [ZKPDL](#) [MEKHL'10] for commitments and NIZKs



# Implementation

---

Used [MIRACL](#) [Scott] for blind IBE, [ZKPDL](#) [MEKHL'10] for commitments and NIZKs

Collected timing information on both a [MacBook Pro](#) (acting as the [TC](#)) and an [ARM v5TE](#) (acting as the [OBU](#))

# Implementation

---

Used [MIRACL](#) [Scott] for blind IBE, [ZKPDL](#) [MEKHL'10] for commitments and NIZKs

Collected timing information on both a [MacBook Pro](#) (acting as the [TC](#)) and an [ARM v5TE](#) (acting as the [OBU](#))

When are blind IBE operations happening?

# Implementation

---

Used **MIRACL** [Scott] for blind IBE, **ZKPDL** [MEKHL'10] for commitments and NIZKs

Collected timing information on both a **MacBook Pro** (acting as the **TC**) and an **ARM v5TE** (acting as the **OBU**)

When are blind IBE operations happening?

- **Encryption**: during Payment process
- **Extraction**: during Audit (OBU as authority, TC as user)
- **Decryption**: during Audit (TC needs to trial decrypt each ciphertext)

# Various measurements: time and space

---

# Various measurements: time and space

---

Operation	Time (ms)	
	Laptop	ARM
Creating parameters	75.12	1083.61
Encryption	82.11	1187.82
Blind extraction (user)	13.13	214.06
Blind extraction (authority)	11.21	175.25
Decryption	78.31	1131.58

Time for blind IBE

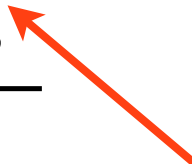
# Various measurements: time and space

---

Operation	Time (ms)	
	Laptop	ARM
Creating parameters	75.12	1083.61
Encryption	82.11	1187.82
Blind extraction (user)	13.13	214.06
Blind extraction (authority)	11.21	175.25
Decryption	78.31	1131.58

Time for blind IBE

cost for OBU during  
Audit is reduced



# Various measurements: time and space

Operation	Time (ms)		Object	Size (B)
	Laptop	ARM		
Creating parameters	75.12	1083.61	NIZK	5455
Encryption	82.11	1187.82	Commitment	130
Blind extraction (user)	13.13	214.06	Ciphertext	366
Blind extraction (authority)	11.21	175.25	Total Pay segment	5955
Decryption	78.31	1131.58	Audit message	494

Time for blind IBE

Size for messages

cost for OBU during  
Audit is reduced



# Various measurements: time and space

Operation	Time (ms)	
	Laptop	ARM
Creating parameters	75.12	1083.61
Encryption	82.11	1187.82
Blind extraction (user)	13.13	214.06
Blind extraction (authority)	11.21	175.25
Decryption	78.31	1131.58

Time for blind IBE

Object	Size (B)
NIZK	5455
Commitment	130
Ciphertext	366
Total Pay segment	5955
Audit message	494

NIZK size dominates total size

Size for messages

cost for OBU during Audit is reduced

# Various measurements: time and space

Operation	Time (ms)		Object	Size (B)
	Laptop	ARM		
Creating parameters	75.12	1083.61	NIZK	5455
Encryption	82.11	1187.82	Commitment	130
Blind extraction (user)	13.13	214.06	Ciphertext	366
Blind extraction (authority)	11.21	175.25	Total Pay segment	5955
Decryption	78.31	1131.58	Audit message	494

NIZK size dominates total size

Size for messages

cost for OBU during Audit is reduced

Time for blind IBE

## Time for TC to perform Audit

Length	Time step	Segments	Time for TC (s)
1 mile	1 minute	2000	55.68
1 mile	1 hour	1000	33.51
2 miles	1 hour	500	10.45

# Various measurements: time and space

Operation	Time (ms)	
	Laptop	ARM
Creating parameters	75.12	1083.61
Encryption	82.11	1187.82
Blind extraction (user)	13.13	214.06
Blind extraction (authority)	11.21	175.25
Decryption	78.31	1131.58

Object	Size (B)
NIZK	5455
Commitment	130
Ciphertext	366
Total Pay segment	5955
Audit message	494

NIZK size dominates total size

Time for blind IBE

time to iterate dominates cost for TC

Time for TC to perform Audit

Size for messages

cost for OBU during Audit is reduced

Length	Time step	Segments	Time for TC (s)
1 mile	1 minute	2000	55.68
1 mile	1 hour	1000	33.51
2 miles	1 hour	500	10.45

# Outline

---

Cryptographic background

Milo

Evaluation

**Conclusions**

# Conclusions

---

# Conclusions

---

We presented [Milo](#), a privacy-preserving electronic toll collection system

# Conclusions

---

We presented **Milo**, a privacy-preserving electronic toll collection system

- Guarantees honesty even in the face of **driver collusion**
- Did so using blind IBE
- Found that computational overhead was manageable, significantly cheaper than certain alternatives



# Conclusions

---

We presented **Milo**, a privacy-preserving electronic toll collection system

- Guarantees honesty even in the face of **driver collusion**
- Did so using blind IBE
- Found that computational overhead was manageable, significantly cheaper than certain alternatives

Future work:

- Possibly formalizing security definitions
- Find **cheaper methods** for achieving same security properties

# Conclusions

---

We presented **Milo**, a privacy-preserving electronic toll collection system

- Guarantees honesty even in the face of **driver collusion**

- Did so using

- Found that it is significantly cheaper than

**Thanks!**  
**Any questions?**

Future work:

- Possibly formalizing security definitions
- Find **cheaper methods** for achieving same security properties