



IEEE International Conference on Blockchain and Cryptocurrency  
27–31 May 2024 // Dublin, Ireland



# zkGen: Policy-to-Circuit Transpiler

**Jan Lauinger**, Jens Ernstberger, Sebastian Steinhorst

Technical University of Munich

TUM Department of Electrical and Computer Engineering

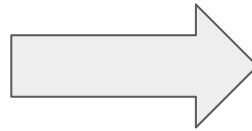
Associate Professorship of Embedded Systems and Internet of Things

Munich, May 2024

# Motivation



Zero-knowledge  
Proof Systems

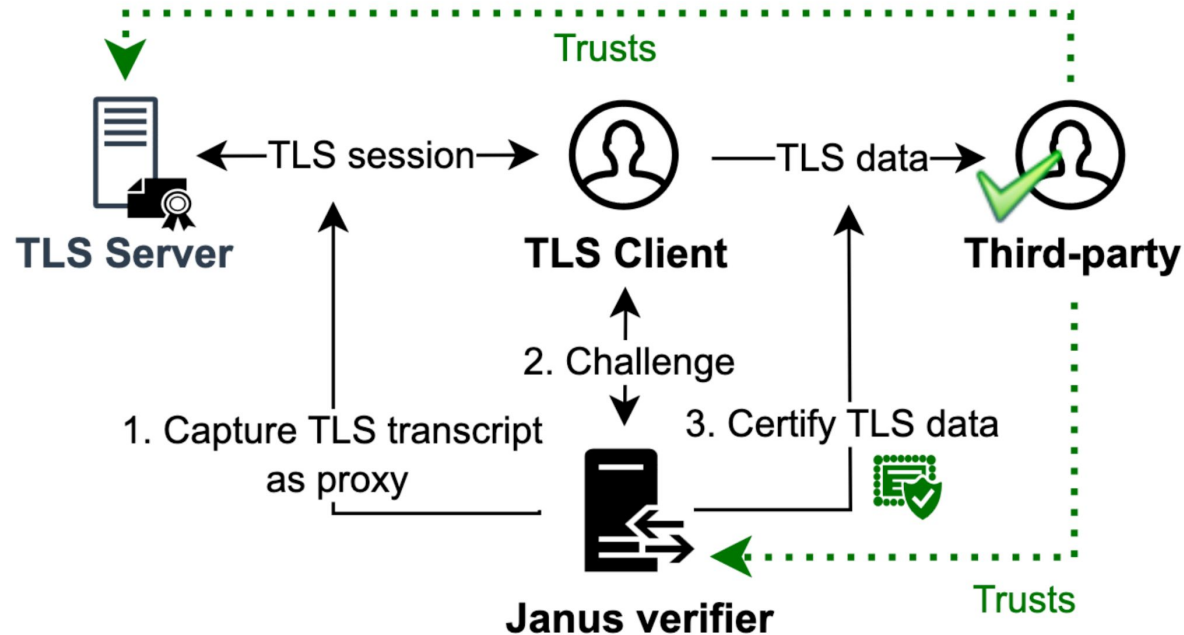


Confidentiality

Compression (SNARKs)

Credibility

# Motivation: Credibility for Data Provenance




# Problem



origo |


Connect Wallet

## Supported Projects



PayPal Sandbox API

+ Connect



Twitter Account API

(coming soon)




Plaid Fintech API

(coming soon)

## Manage Passport

Passport Data:

Extension: (MetaMask) - MetaMask



**Welcome back!**  
The decentralized web awaits

Password

Unlock

Close x



Value:

Constraint:

More Information

Verify

# Problem



“paypal bank balance > 100\$”

```
commit(r, secret) == c
k = HKDF(secret)
plaintext = decrypt(k, ciphertext)
substring = find(plaintext, pattern)
bank_balance = convert(substring)
bank_balance > 100
```

A simple **statement**

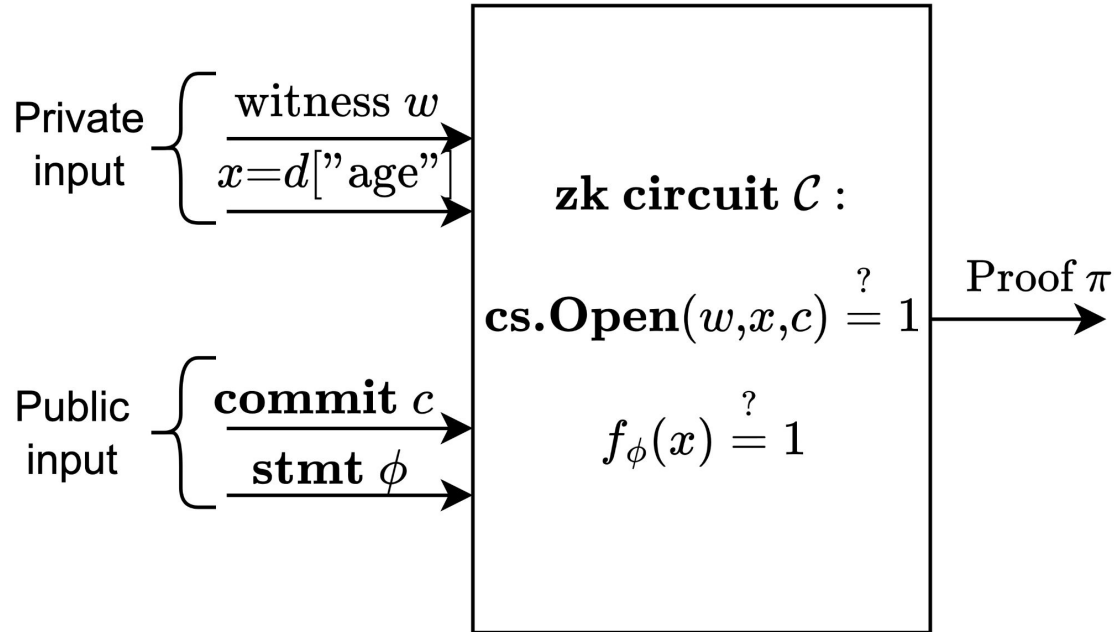
A complex **circuit** (#constraints=1.6million)

# Contributions



- zkPolicy Language
- Transpiler Architecture
  - input: zkPolicy
  - output: ZKP circuit expressed in domain-specific language (DSL)
- Reducing the **description complexity** of privacy-preserving computations

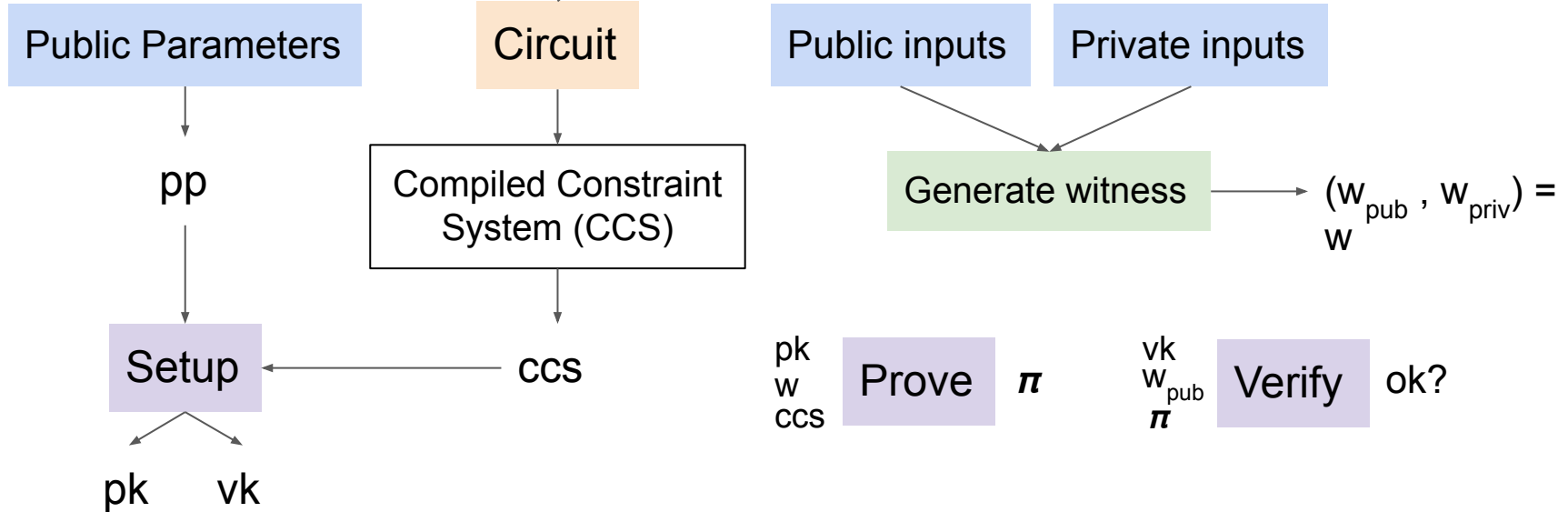
# Background: Zero-knowledge Systems



# Background: Zero-knowledge Systems

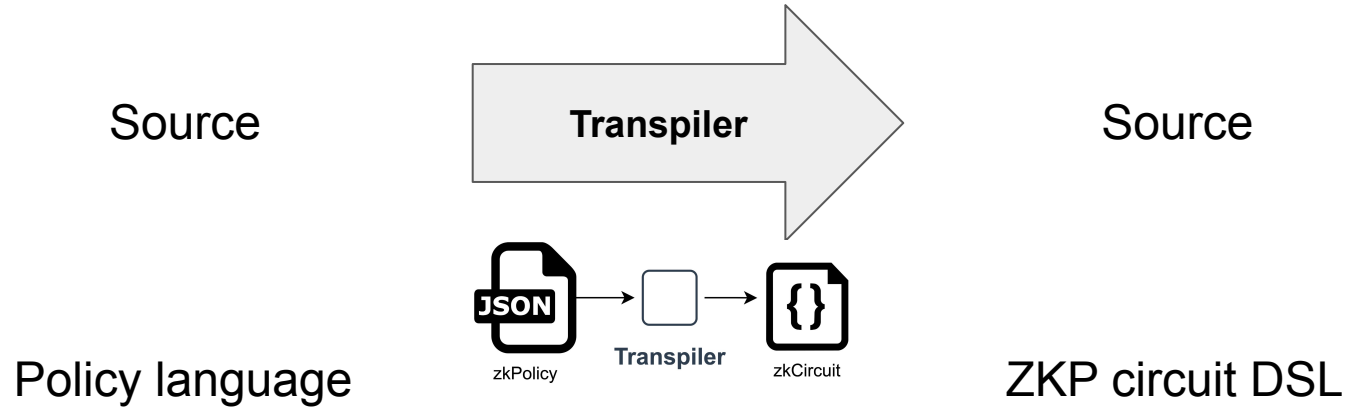


written in domain-specific language (DSL)

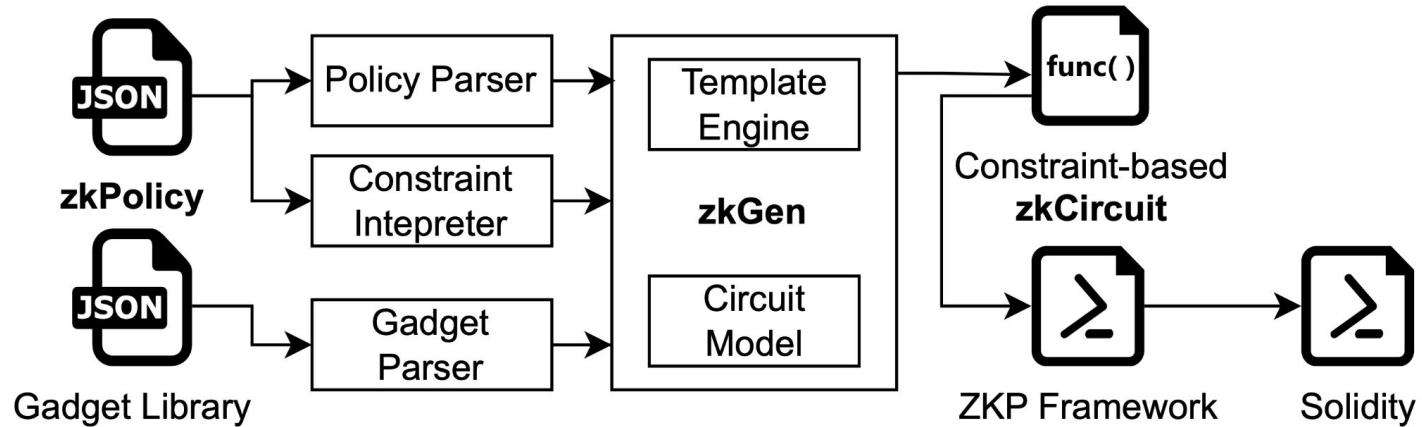




# Background: Transpiler



# Approach: zkGen Transpiler Architecture



# Approach: zkPolicy Language & Gadget Library



```
zkpolicy > {} example3_zkFriendlyCommitData.json > ...
1  {
2    "name": "AgeCommitCheck",
3    "relations": [
4      {
5        "private_argument": {
6          "name": "Age",
7          "type": "string",
8          "protection": {
9            "algorithm": "commitments:mimc",
10           "parameter": "message"
11         }
12       },
13       "public_argument": {
14         "name": "Threshold",
15         "type": "integer"
16       }
17     }
18   ],
19   "constraints": [
20     "0:age-gt-0:threshold"
21   ]
22 }
```

```
1  {
2    "commitments": {
3      "mimc": {
4        "commit_string": {
5          "scope": "public",
6          "type": "string",
7          "size": 32
8        },
9        "witness": {
10         "scope": "private",
11         "type": "string",
12         "size": -1
13       },
14       "message": {
15         "scope": "private",
16         "type": "string",
17         "size": -1
18       }
19     },

```

# Evaluation: Transpilation Results



<b>zkPolicy</b>	<b>LOC policy</b>	<b>LOC circuit</b>	<b>Transpile time</b>
TLS commit	22	975	2.38 ms
Age commit	20	85	1.70 ms

# Discussion: Limitations & Future Work

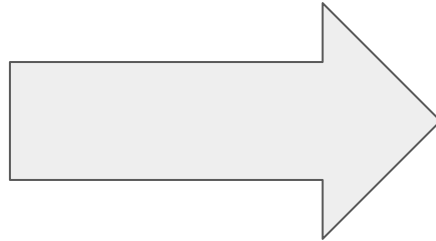


- ZKP DSL only, no coverage of MPC or HE DSLs
- Integrate formal verification of generated circuits
  - detect underconstrained circuits when gadget compositions are used

# Conclusion



- zkPolicy Language
- zkGen Transpiler



**Concise description** of compliant and private computations with a **fraction** of code lines.

Questions?

