Shattered Chain of Trust: Understanding Security Risks in Cross-Cloud IoT Access Delegation

Bin Yuan, Yan Jia, Luyi Xing,

Dongfang Zhao, Xiaofeng Wang, Deqing Zou, Hai Jin, Yuqing Zhang

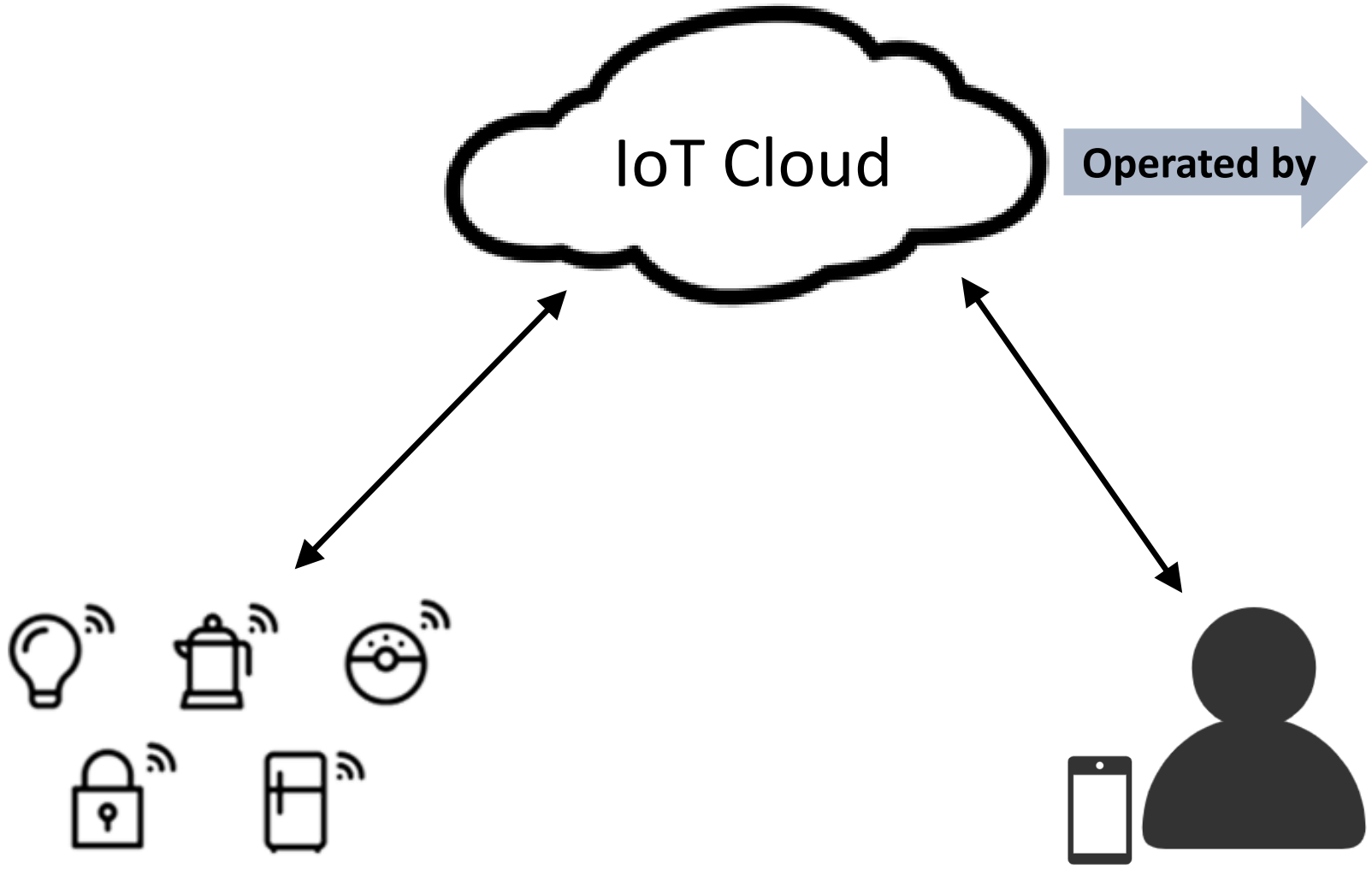
華中科技大學 Huazhong University of Science and Technology

INDIANA UNIVERSITY BLOOMINGTON

西安電子科技大學 XIDIAN UNIVERSITY

中国科学院大学 University of Chinese Academy of Sciences

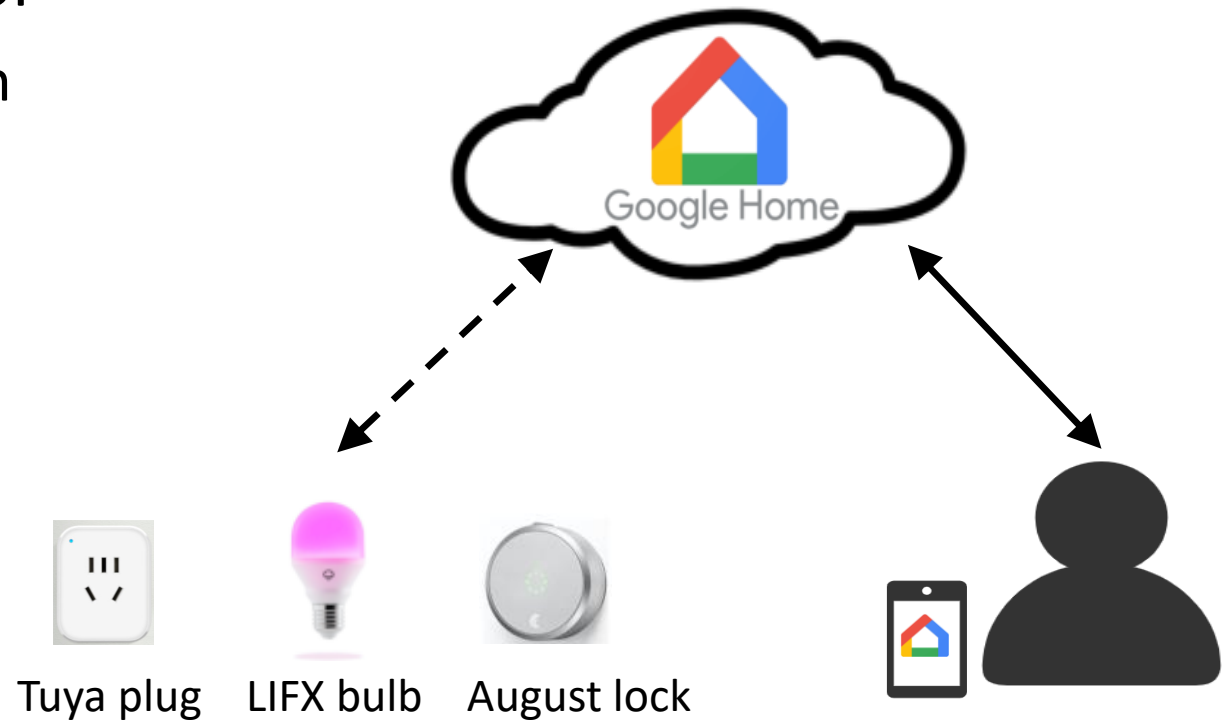
The Cloud-based IoT Access



- IoT device vendors
  - Philips, August, LIFX, etc.
- Cloud service providers
  - Google, Amazon, IFTTT, etc.

Cross-Cloud IoT Device Access

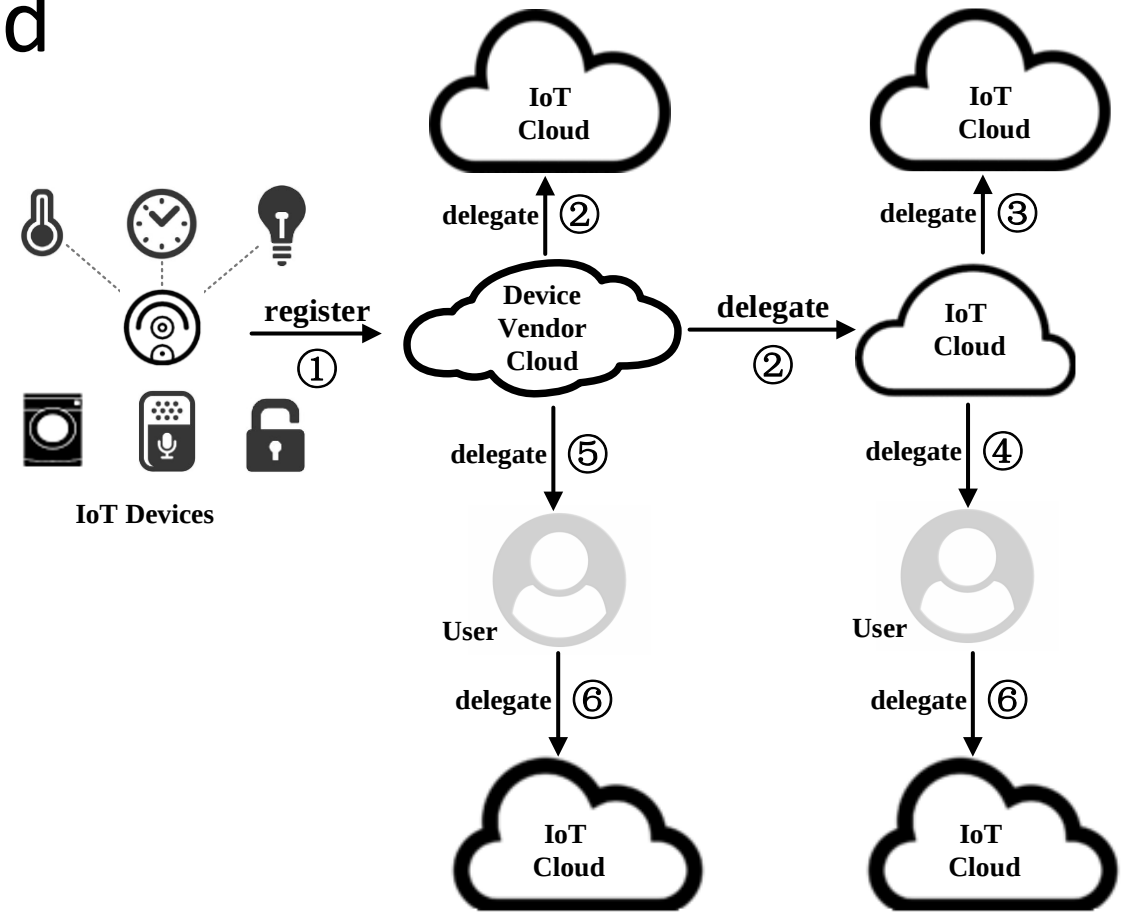
- Cross-vendor/cross-cloud device control
  - Manage different vendor devices through the same console
- Sharing of device access
  - Share the access to the lock to an Airbnb guest (temporarily)



A user uses Google Home to control all her devices from different vendors (e.g., Tuya plug, LIFX bulb, August lock)

Cross-Cloud IoT Device Access Delegation

- Delegation mechanisms in the wild
  - OAuth and its customization
    - Actions on Google
  - Custom authorization
    - IFTTT & SmartThings
- Convoluted delegation chain
  - across different clouds and users



Threat Model & Security Requirements

- Threat model
  - Delegatee user can be malicious, while the administrator, cloud, and device are benign
  - Goal of the adversary is to get unauthorized access to IoT devices
  - The adversary would make full use of his power to acquire useful information, e.g., make API calls, extract information from system logs, official documentations and capture network traffic generated by/for his mobile app

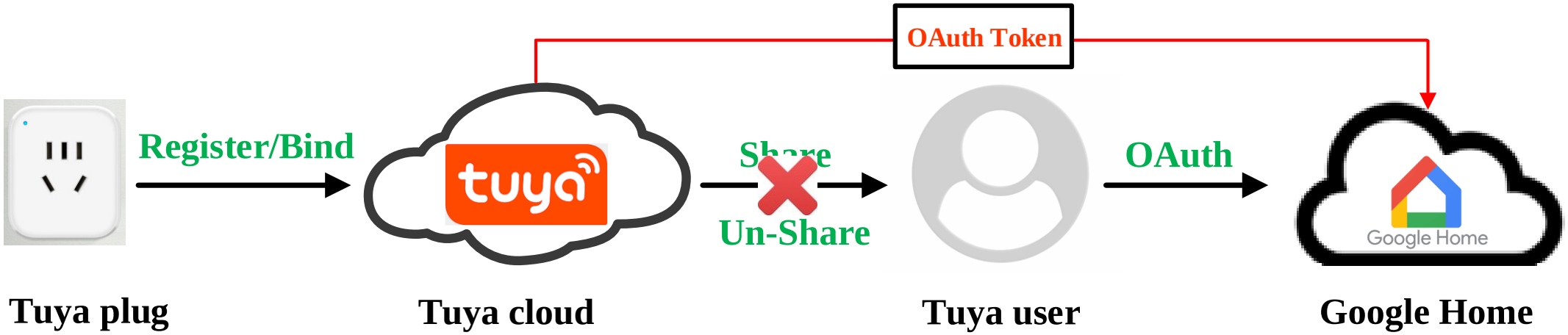
- Security requirements
  - Safe and consistent delegation policies
  - Non-bypassable and transitive delegation control

Risks in Cross-Cloud Delegation

- Theoretic models analyzed before
  - all parties run the same delegation protocol and interact through unified interfaces
- Delegation in today's real-world IoT clouds
  - individual, heterogeneous delegation protocols
  - incompatible with other clouds
  - not being properly verified

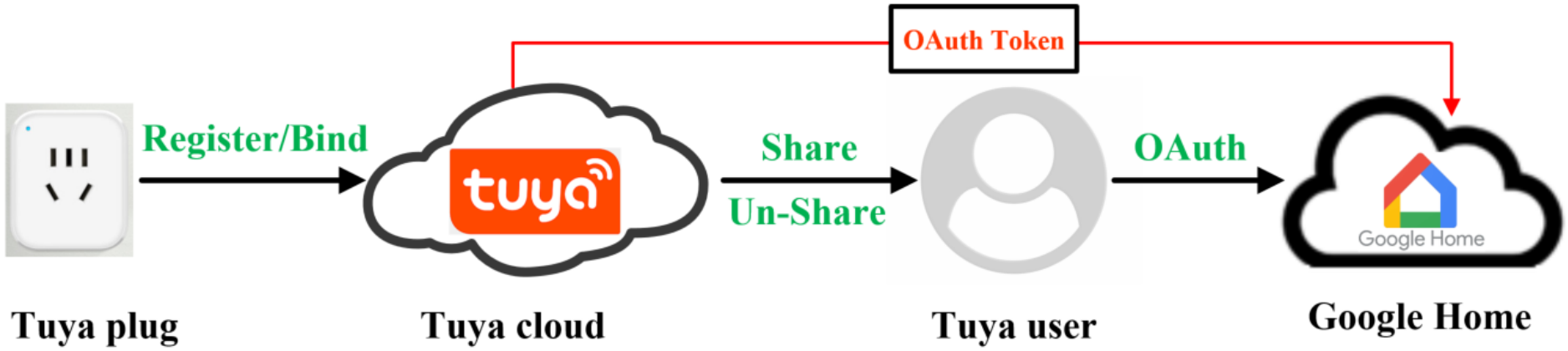
Vulnerable Cross-Cloud IoT Delegation: a motivating example

- Violation of “transitive delegation control” in Tuya



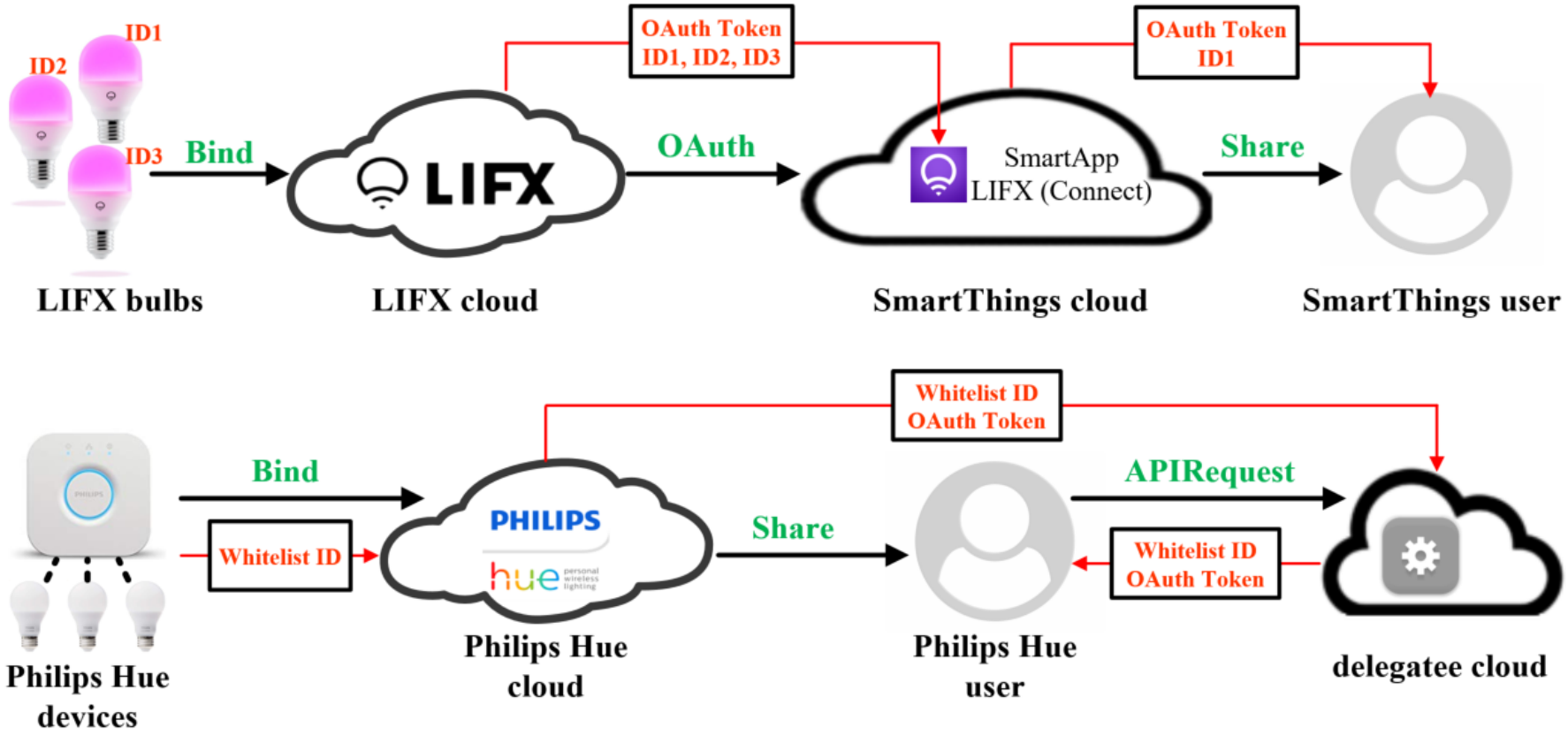
  - Google Home still holds a valid OAuth token (issued by the Tuya cloud during the OAuth operation), allowing Google Home to access the Tuya plug even after the un-share operation

Observations from the Tuya Case



- Multiple operations supported in an IoT setting
- Data flow (e.g., token issuing and distributing) along with operations
- Access control check and access request forwarding
- Multi-step access path (with a valid token)

Towards Vulnerability Discovery More Automatically



Common delegation pattern identified in different settings

Formal verification based IoT cross-cloud delegation vulnerability discovery

VerioT: the first (semi-automatic) verification tool for IoT cross-cloud delegation vulnerability discovery

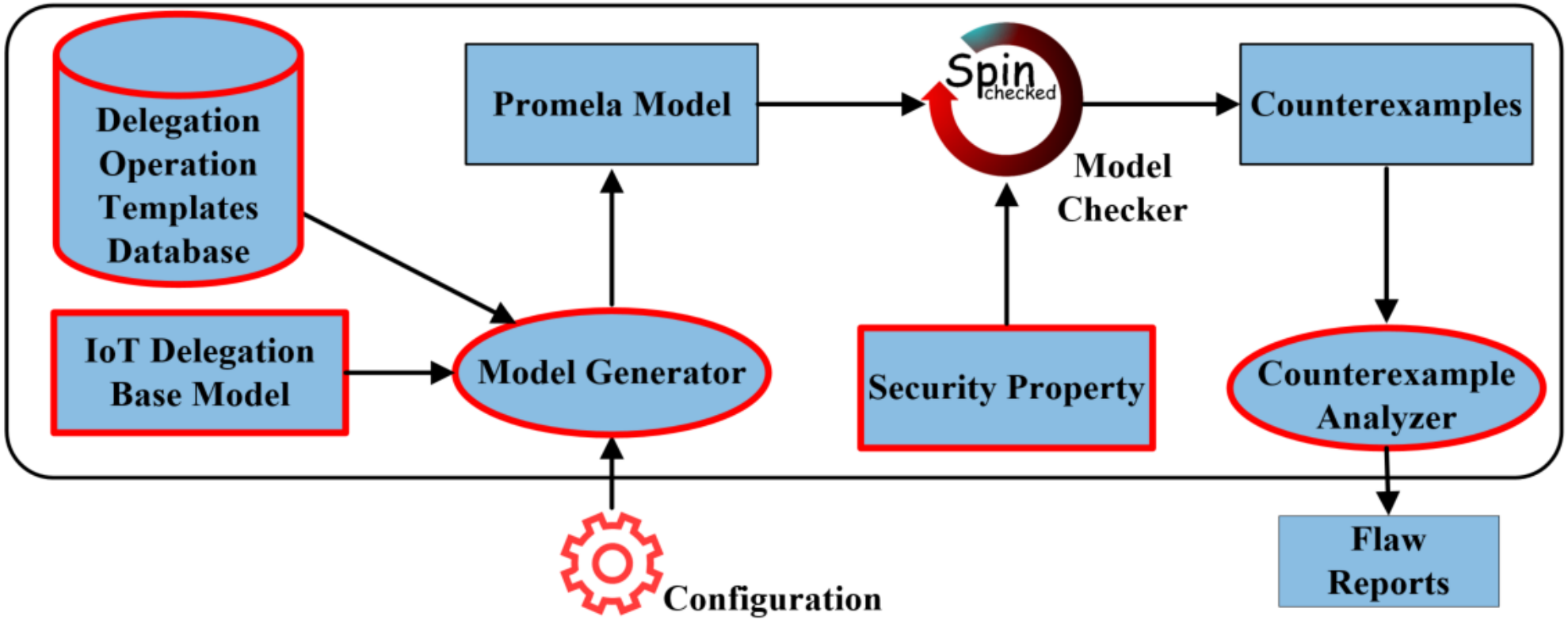
- Security property
  - unauthorized delegatee user should not have a path to the IoT devices which he is not entitled to access

- IoT delegation modeled as a transition system $\mathcal{M} = (\mathcal{A}, \mathcal{S}, O, \mathcal{T}, s_0)$
  - $\mathcal{A}$ is the set of **actor**s (e.g., device, cloud, user)
  - O is the set of **operation**s (e.g., OAuth, share, un-share, bind, unbind, APIRequest, etc.)
  - $\mathcal{S}$ is the set of **state**s, where s_0 is the **initial state** (where no delegation happens)
    - Tokens received and issued during delegation and the access control mapping between these tokens
  - $\mathcal{T}: \mathcal{S} \times O \rightarrow \mathcal{S}$ is a function that drives the **transition** from one state to the next

- Detecting flaws
  - leveraging a model checker to verify whether pre-defined security properties hold in the model

VerioT: the first (semi-automatic) verification tool for IoT cross-cloud delegation vulnerability discovery

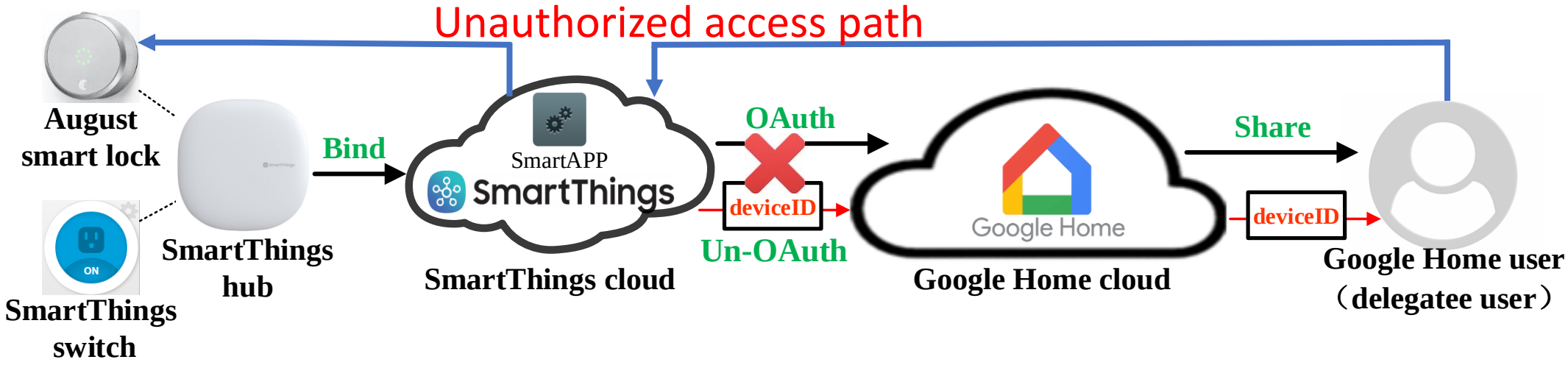
- Modeling different real-world IoT system
  - **Refinement**: base model, operation template, and configuration



The architecture of VerioT

Findings 1: Inadequate Cross-Cloud Coordination

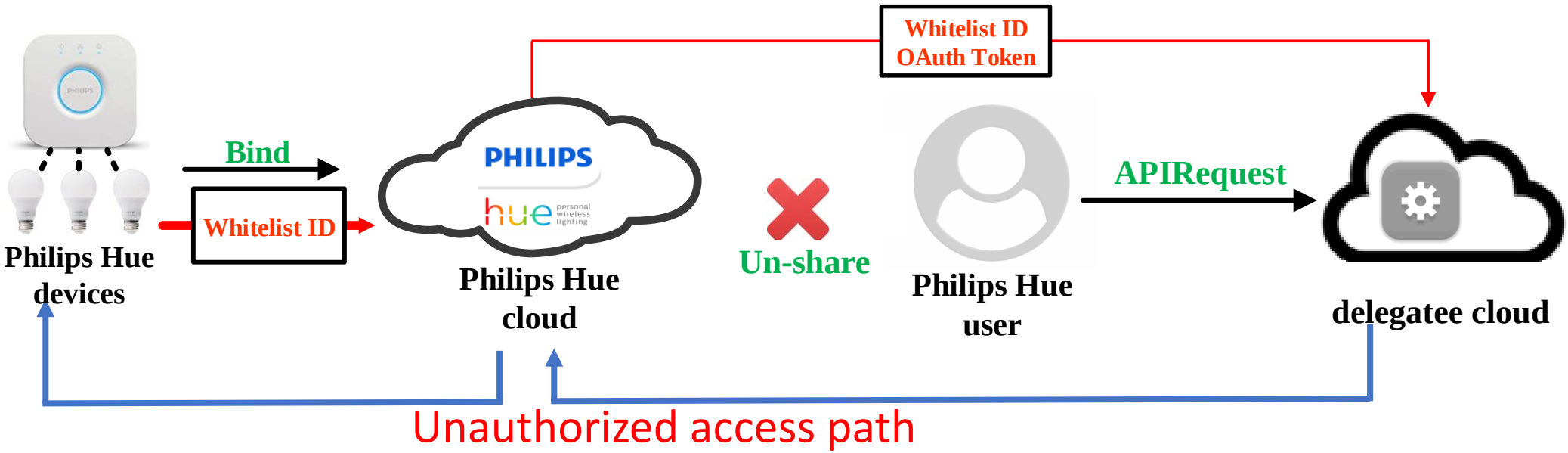
- Mis-aligned security requirements between clouds
  - Google's lack of knowledge about the security implication of SmartThings' device ID



- Malicious delegatee user (e.g., an Airbnb guest) can use the device ID to spoof events to trigger SmartThings to open the lock even after he checks out

Findings 2: Inadequate Policy Enforcement

- Incomplete revocation scheme in the delegator cloud
  - Philips Hue cloud only invalidates the token which is used for access check in the device, not the token that is used for authentication in the cloud



- Malicious delegatee user can abuse the API to regain remote access to the Philips Hue devices even after the administrator revokes his access right

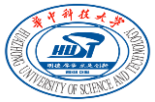
Conclusion

- Root cause
  - Heterogeneous and ad-hoc delegation process (because of the absence of a standardized, fully verified cross-cloud delegation protocol)
- Lessons learnt
  - The caution one should take when applying a custom cross-cloud authorization scheme to today's already complicated IoT delegation
    - the delegator and the delegatee violate each other's security policies
    - problematic security policy enforcement due to lack of rigorous verification
- New design principles
  - Communicating security assumptions and constraints
  - Decoupling the delegatee and the delegator clouds
  - Verifying delegation design whenever possible

Thanks!

Please refer to our website for the source code of our tool, the vendor response to our responsible disclosure, the PoC attack demos and the full list of affected vendors

https://sites.google.com/view/shattered-chain-of-trust-under/home?authuser=0


華中科技大學
Huazhong University of Science and Technology


INDIANA UNIVERSITY
BLOOMINGTON


西安電子科技大學
XIDIAN UNIVERSITY


中国科学院大学
University of Chinese Academy of Sciences