DUST-BT: Preventing Supply Chain Tampering using Blockchain Technology

Aniket Kate, Mahimna Kelkar, Easwar Vivek Mangipudi, Pedro Moreno-Sanchez, Krutarth Rao
Purdue University

(1) Supply Chain

**Purpose:** Supervised transfer of products from suppliers to consumers

Traditional Supply Chains:
- Track product components using RFID
- Local logs maintained at each supplier

(2) What is the Problem?

**Counterfeiting.** How to distinguish between products with genuine components and counterfeit ones?

Global trade in fake goods worth nearly half a trillion dollars a year!

(3) Building Block: Blockchain

- **Interesting properties:**
  - Conflicting transactions can be easily detectable checking the blockchain
  - Transactions added to the blockchain cannot be removed
- **Successfully deployed** in several applications today:
  - Cryptocurrencies (Bitcoin, Ethereum)
  - Credit networks (Ripple, Stellar)

**Question:** Can we leverage blockchain to avoid counterfeit in supply chain?

**Challenges:**
1. Blockchain must maintain transactions tailored to supply chain
2. Complex logic for product management

(4) Our Solution: DUST-BT

1. **Create smart contract:** All supply chain participants agree on a set of rules and logic
2. **Deploy smart contract:** Rules and logic are installed in a set of validators. Validators initialize the blockchain
3. **Append transactions to blockchain:** Transactions (e.g., create or transfer product) are added to the blockchain according to the smart contract

(5) DUST-BT: Smart Contracts

**Main ideas:**
- Convert agreed logic and rules into functions
- Automatic verification of supply chain correctness

```
NewProduct(Product p)
if p \notin blockchain then
  blockchain.insert(p)
return OK
else
  return ERROR

TransferProduct(Product p, CurrentEntity ce, NewEntity ne)
if p \in blockchain then
  if p.currentEntity = ce then
    p.currentEntity = ne
  blockchain.update(p)
return OK
return ERROR

TransferProductSignal(Product p, CurrentEntity ce, NewEntity ne)
if p \in blockchain then
  if p.currentEntity = ce \land externalSignal() = True then
    p.currentEntity = ne
  blockchain.update(p)
return OK
return ERROR
```

(6) DUST-BT: Implementation

Our implementation is composed of:
- **Hyperledger architecture:** Hyperledger software
- **Smart Contracts:** Our own implementation in Golang
- **Graphical Interface:** Our own implementation in Flask and Javascript

(7) Conclusions

- Counterfeiting can be prevented leveraging blockchain technology
- DUST-BT offers a flexible yet effective supply chain management

**Acknowledgments:** This project is supported by Northrop Grumman