Distributed Ledger Technology (DLT):

On breaking DLT-based Ecosystems

Reza Hedayat

Head of Security Innovation

infoGuard
SWISS CYBER SECURITY

6th ISG Alumni Conference
Royal Holloway, University of London

June 25, 2018

Content

Introduction

WhoAmI I

- **Currently@InfoGuard**
  - Security services for emerging technologies (IoT, DLT, ...)
  - Security Research Lab (Support RED & BLUE Team)
- **Previously**
  - @FLYNT Bank AG
    - Sr. Security Architect
  - @AdNovum
    - Software Security Engineer
    - Security Consultant
    - Security Architect

WhoAmI II

- **Background**
  - Computer Science
    - Software Engineering
    - Cryptology
    - Neuronal Networks / Fuzzy Logic
  - Information Security
    - Cryptography
    - Smart Cards / Tokens
    - Malware

WhoAmI III

Passion

- ♡ Cryptography ♡
- Malware and its Underground Economy
- Full-Stack Exploit Engineering
- Systems Security

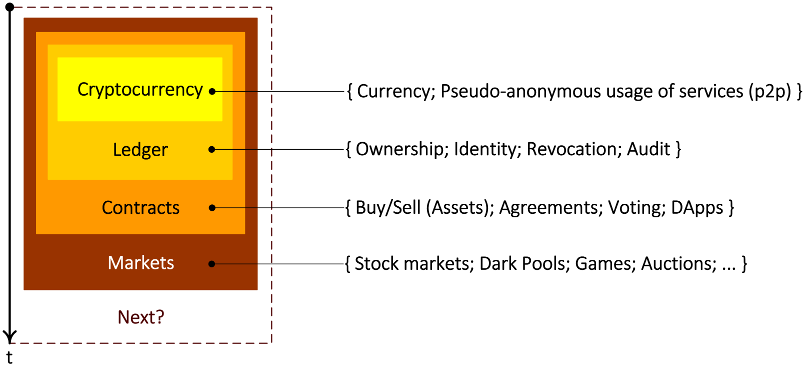
DLT Defintions - An attempt I

- From a **computer science** perspective
  - A **deterministic state machine** with two **main functionalities**:
    1. A globally accessible state (Singleton)
    2. A virtual machine that is able to change this state
- From a **practical** perspective
  - A **world-computer**
  - A **globally decentralised computing infrastructure**, that runs **programs** (Smart Contracts)

DLT Defintions - An attempt II

- Integrates an **economic function**
  - Every usage of a resource costs $\rightarrow$ cryptocurrency
- Enables decentralised applications that **reduce** censor, interfaces of third parties and thus **counterparty risk**

DLT Generations

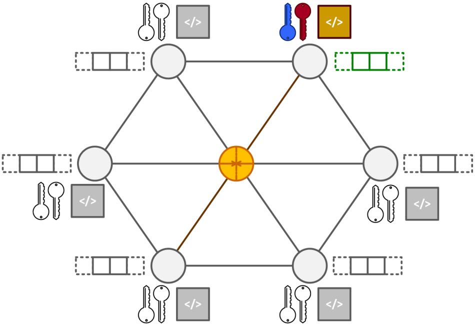


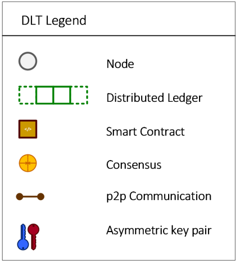
DLT as Panacea

- No matter in which line of business
  - Insurance
  - Banking
  - Real estate
  - Governance
  - ...
- Is prophesied for everything that should be somehow **valid**

Thanos' reaction to DLT security claims

DLT Anatomy

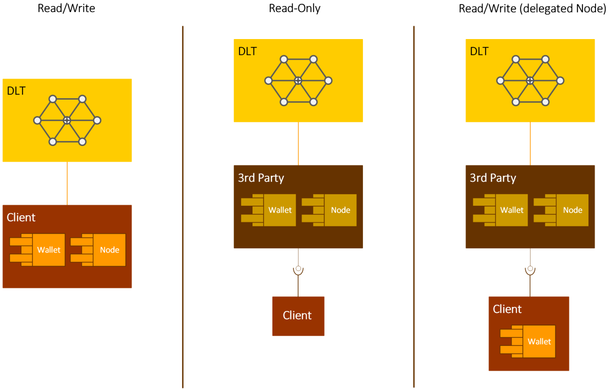

</>
</>
</>
</>
</>
</>


DLT Legend
Node
Distributed Ledger
Smart Contract
Consensus
p2p Communication
Asymmetric key pair

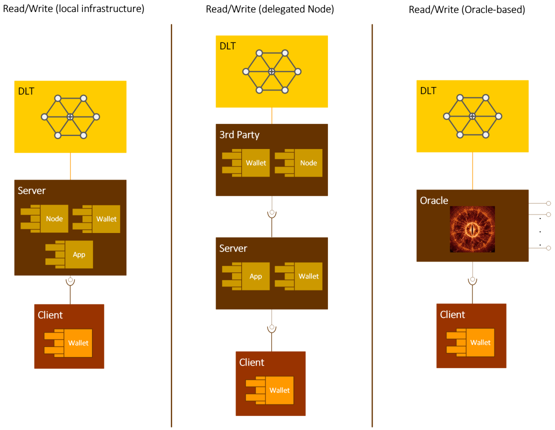
DLT security properties

- Confidentiality $\rightarrow$ **not given by default**
- Non-repudiation $\rightarrow$ **digital signatures**
  - Integrity
  - Data origin authentication
- Availability $\rightarrow$ **p2p**
- Agreement/Double Spending $\rightarrow$ **Consensus**

Serverless DLT

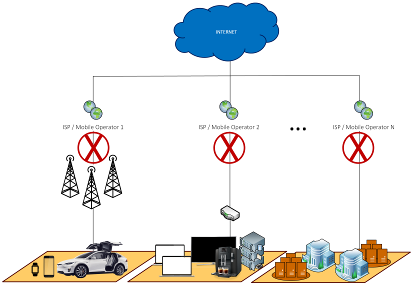


Server-based DLT



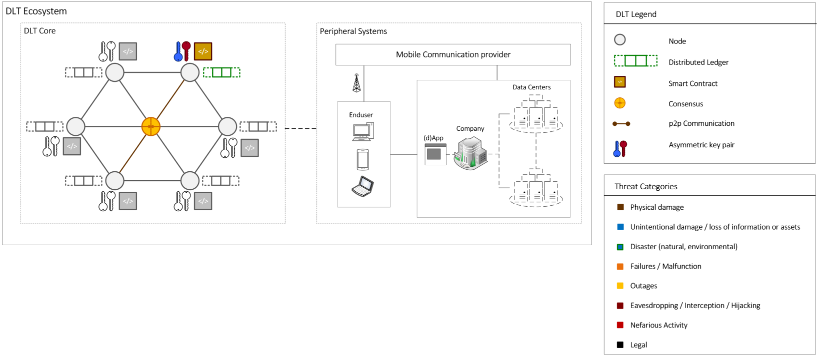
Convergence: Total decentralisation

- Elimination of all central nodes (e.g. ISPs, Operators, ...)
- WMN-based communication (Wireless Mesh Network)
    - Example: RightMesh and the right to be connected [7]
- **Re-balancing might and power** → fair society?

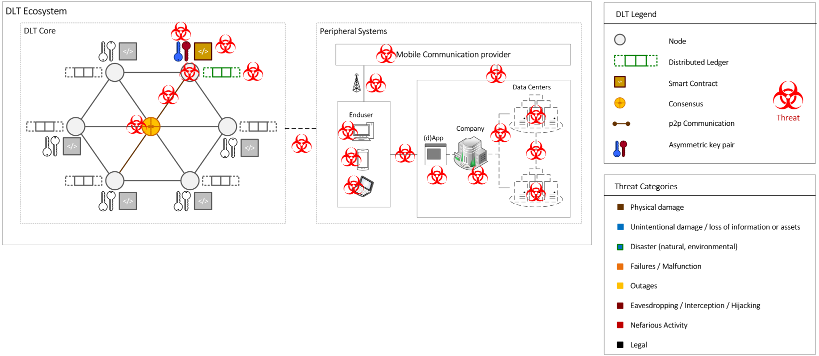


Threat Landscape

Overview of DLT Threats I



Overview of DLT Threats II



Consensus & p2p Communication

Consensus	Voting-approach	Threats
Virtual Voting	Loyal nodes; Transparency	Impractical
PoW	Machine power	$f < \frac{1}{3}$; Eclipse
PoS	Wealth	Nothing at Stake
Gossip	Random communication	$f < \frac{1}{3}$; Centralised; Closed Source;
DPoS	Delegation	Partially centralised
PoA	Admins	Centralised
...	...	...

$*f :=$ malicious node

Ledger

- **Leakage**
  - Transparency facilitates the reconnaissance phase (see cryptography example)
  - Data privacy implications $\rightarrow$ GDPR
- **Sidechains**

Cryptography I

- **Design flaws**
  - Standards
  - Customized
  - Back doors
- **Implementation errors**
  - Arithmetic core
  - Algorithm
    - Service: Encryption, Signature, ...
  - Scheme
    - Parsing, input and output validation, encoding, ...
  - Parametrisation
  - Key management
  - ...

Cryptography II

- **Example: ECDSA**
  - **Signature generation**
    1. Generate an ephemeral key k_E with $0 < k_E < q$ at random
    2. Compute $R = k_E A$
    3. Let $r = x_R$
    4. Compute $s \equiv (h(x) + d \cdot r)k_E^{-1}$ *mod* q.
    5. Return the signature (r, s)
  - **Attack**
    1. Monitor all transactions on the ledger
    2. Extract r from the signature and check if r is re-used
    3. If yes $\rightarrow k = \frac{h(m_1) - h(m_2)}{s_1 - s_2}$ *mod* q and $d \equiv (sk - h(m))r^{-1}$ *mod* p

Cryptography III

- **Example: zkSNARKS**
  - Structure:

$$< \textbf{encryptedData} > || < \textbf{proof} >$$

  - **Highly sensitive key ceremony**
    $\rightarrow$ Leakage is detrimental $\rightarrow$ forging proofs
  - **Is not resistant** to **quantum computers**

Nodes

- **Trusted Computing Base (TCB)**
  - Hardware, Firmware, OS
- **Wallet/App**
  - Password strength
  - Implementation errors
  - Vulnerabilities in used libraries (e.g. Node.js, Meteor, ...)
  - ...
- **Ledger API**

Smart Contracts

- **Design- and implementation errors**
  - Initialization
  - Logic flow
  - Calculation
  - Boundary condition violations
  - Parameter passing
  - Input validation and output encoding
  - Resource exhaustion
  - Race condition
  - ...

Concrete attacks against Ethereum

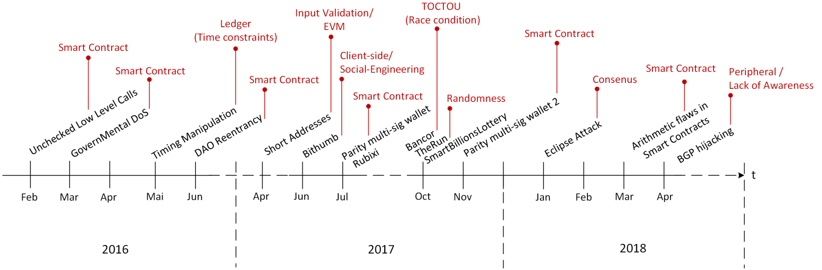
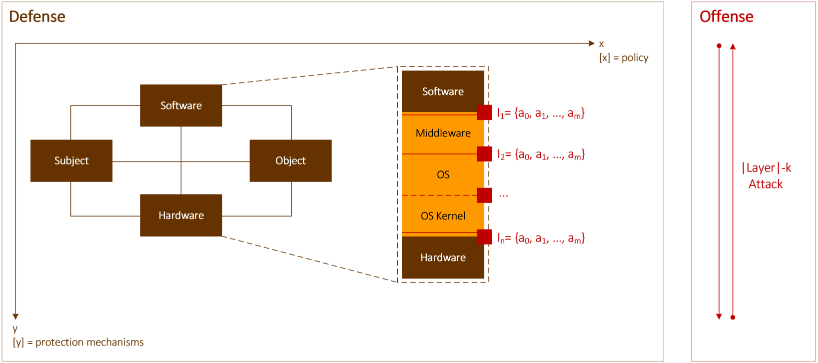


Figure: (Sources: [2, 4])

Mitigation

Trust I


Defense
x
[x] = policy
y
[y] = protection mechanisms
Software
Subject
Object
Hardware
Software
Middleware
OS
OS Kernel
Hardware
$I_1 = \{a_0, a_1, \ldots, a_m\}$
$I_2 = \{a_0, a_1, \ldots, a_m\}$
...
$I_n = \{a_0, a_1, \ldots, a_m\}$
Offense
|Layer|-k
Attack

Trust II

Example: HSM/SE/TPM

- Security goals
  - Secure generation, usage and storage of cryptographic keys
  - Secure execution of cryptographic operations
- Assumptions
  - Tamper-resistance
  - Strong RNG (unbiased)
  - No leakage (anti-side-channels)
  - Proper implementation of interfaces (e.g. PKCS#11, JCE)
  - ...

Trust III

Example: ASLR (Address Space Layout Randomization)

- Security goal
  - Increasing the difficulty of predicitng the memory layout of a process
- Assumptions
  - High entropy
  - Strong RNG (unbiased)
  - ...

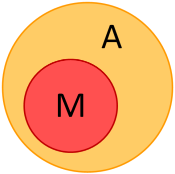
Trust IV

Example: Memory Isolation / CPU bounds (Meltdown/Spectre)

- Security goals
  - Separation of kernel- and user-space
  - CPU executes all instructions correctly
- Assumptions
  - Proper implementation of *out-of-order execution* (Meltdown [6])
  - Proper implementation of *speculative execution* (Spectre [5])

Malware Problem

- $\mathcal{A}$: The set of all programs
- $\mathcal{M} \subseteq \mathcal{A}$: The set of all malware
- $D_{\mathcal{M}}$: A perfect malware detector
- $m \in \mathcal{M}$: A malware instance



Proof. (Cohen, 1986 [3])

1. $D_{\mathcal{M}}(m) = \top$ (Tautology)
2. $D_{\mathcal{M}}(m) = \bot$ (Contradiction)

$\implies$ If there was a perfect malware detector $D_{\mathcal{M}}$, it could also solve the **Halting Problem**.

Malware Success Factors

1. **Not detectable** in general
2. No definition of **malicious behaviour**
3. Software is **full of bugs**
4. **Patch-and-penetrate** approach
   - Life expectation of an exploit on average $\sim$ **7 years** after initial discovery [1]
5. **Obfuscation** techniques
6. Lack of **user awareness**

Recommendations

- Gain **knowledge**
  - DLT fundamentals
  - Security awareness
- **Reduction** of the attack surface
  - Architecture reviews
  - Hands-on security testing
    - Pen-testing and attack simulation (**RED**-Teaming)
    - Static and dynamic analysis (Smart Contracts)
- Gain **reactive capabilities**
  - **BLUE**-Teaming
  - **PURPLE**-Teaming
  - **Threat**-Hunting

Conclusion

DLT as a Technology

- Paradigm shift (Anti-Cloud)
- Promising alternative with regards to known architecture approaches
  - Does it converge to **total decentralisation**?
  - The **fair society**
- **Does not solve** our **core problems** in security
- As a **dual-use technology** perfectly suitable for providing **Malware-as-a-Services (MaaS)**
  - Customer → Smart Contract → Victim

Evolution of Digital Identities

- Pure DLT-based solution shifts the security to the enduser
  - Highly problematic with the current design of security mechanisms
  - **High degree of user awareness** is inevitable!

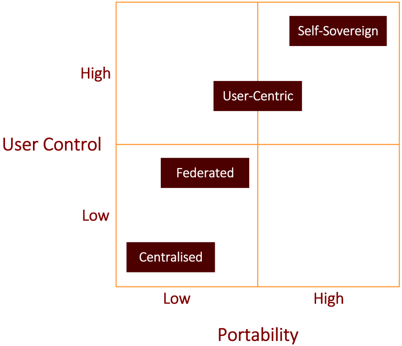


Figure: (Source: [8])

Questions & Contact

Reza Hedayat
Head of Security Innovation
reza.hedayat@infoguard.ch

infoGuard
SWISS CYBER SECURITY

?

Appendix

References I

Lilian Ablon and Andy Bogart. *Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits*. Available at https://www.rand.org/content/dam/rand/pubs/research_reports/RR1700/RR1751/RAND_RR1751.pdf. 2017.

Applicature. *History of Ethereum Security Vulnerabilities, Hacks and Their Fixes*. Available at https://applicature.com/blog/history-of-ethereum-security-vulnerabilities-hacks-and-their-fixes. 2018.

References II

- Fred Cohen. "Computer Viruses". PhD thesis. University of Southern California, Jan. 1986.
- NCC Group. *Decentralized Application Security Project Top 10 2018*. Available at `https://www.dasp.co`. 2018.
- Paul Kocher et al. "Spectre Attacks: Exploiting Speculative Execution". In: *ArXiv e-prints* (Jan. 2018). arXiv: `1801.01203`.
- Moritz Lipp et al. "Meltdown". In: *ArXiv e-prints* (Jan. 2018). arXiv: `1801.01207`.

References III

- RightMesh. *RightMesh - A Decentralized Mobile Mesh Networking Platform Powered by Blockchain Technology and Tokenization*. Available at `https://www.rightmesh.io/`. 2018.
- Andrew Tobin and Drummond Reed. *White Paper: The Inevitable Rise of Self-Sovereign Identity*. Available at `https://sovrin.org/wp-content/uploads/2017/07/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf`. 2016.

Pictures

- Falling Cards → **URL**
- House of Cards → **URL**
- Snake Oil → **URL**
- Thanos → **URL**