WP1
Security, Trust and Privacy

WP1 Status

"Security, trust and privacy"

- Timestamping and Blockchain immutability from computational hardness (non-PoW)
- Short cryptographic proofs of predicates
- Cryptographic mechanisms to share data privately
- Identity management, authentication and authorization

- 4y PhD, Aron van Baarsen, 1-3-2020
- (PhD, Esteban Landerreche)

Research output:

- *[LSS20] Non-interactive Cryptographic Timestamping based on Verifiable Delay Functions*, Financial Crypto 2020
- *[LS18] On Immutability of Blockchains*, ERCIM Workshop 2018

WP1 Research Output:

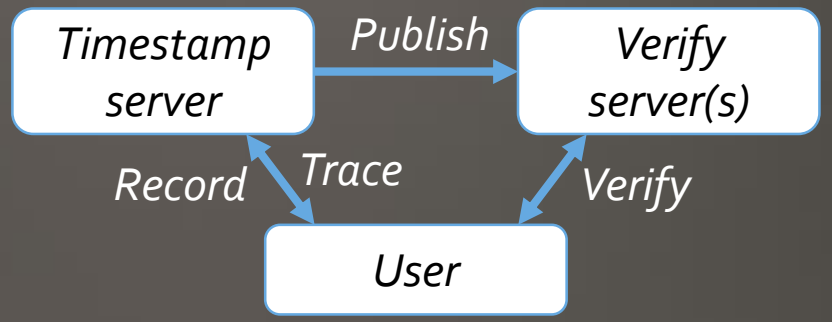
Cryptographic Timestamping

Cryptographic Timestamping

Interactive Timestamping:

- Timestamp server + Verify server
- Record a message *M* at time *Tr*
  - Output receipt to User
  - Output commitment to Verify server
- Verify timestamp of *M* at time *Tp*
  - Timestamp server outputs trace
  - User verifies against Verify server

- Security notions:
  - Backdating security:
    Cannot create claim with *Tf<Tr*
  - Postdating security:
    Cannot create claim with *Tf>Tr*



- [HS91]
  - Use hash-chains: h_0, h_1, …
  - h_i=Hash(h_{i-1}, M_i)
  - Publish all hashes in trusted media
  - Interactive:
    Claim must be verified against media
  - Secure if Prover and Verifier don't collude
- [BHS93]:
  - Batch messages in blocks
    using Merkle Tree

Blockchain Immutability & Timestamping

- Distributed Ledgers (DLT) should be immutable
  - Sufficiently old Transactions cannot be removed, altered, or inserted
  - [Nako8] Bitcoin construction:
    - Hash-chains: any change in a block M_i affects all subsequent hashes
    - Proof-of-Work: for each block solve *parallel computational problem*
  - [Nako8] Immutable against attackers with <50% computational power

- [GMG15]: Timestamping service build on Bitcoin
- [LS18,LSS20]: (Non-interactive) timestamping based on *serial computational problem*

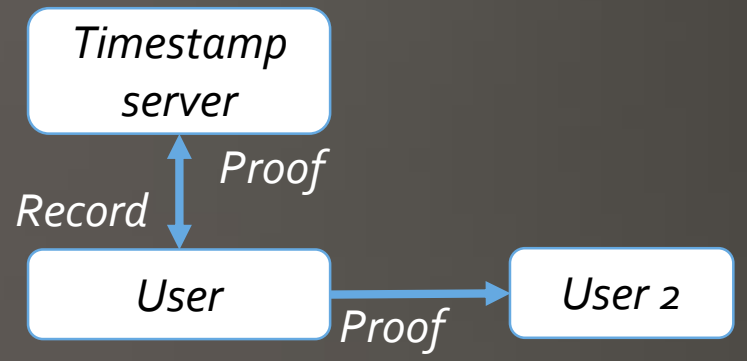
Medium-hard Computational Problems for Timestamping

- Proof-of-Work
  - Input block *M*, difficulty *D*
  - Find *N* such that Hash(*M*)<*D*
  - Massively parallel
  - Bitcoin: 2^{67} H/sec, 2^{91} H/year
  - Massively wasteful: ~65 TWh/year
  - Comparable to power use of Czech Republic
  - Can be used for Bitcoin consensus

- VDF: Verifiable Delay Function
  - Input block *M*, iterations *T*
  - Computes *T* serial operations in *T* wallclock time
  - Generates short efficient proof of correctness
  - Inherently sequential
  - Attacks require much higher speed than honest parties
  - Physical limitations on iterations/sec
  - Need optimized VDF cores for honest usage
  - Can't be used for Bitcoin consensus

(Non-interactive) timestamping using VDFs

- Non-interactive Timestamping:
  - Only need Trusted Timestamping server (no Verify server needed)



  - Record a message *M* at time *Tr*
    - Output receipt to User
  - Verify timestamp of *M* at time *Tp*
    - Timestamp server outputs non-interactive proof of age *(Tp-Tr)*
    - User verifies proof

  - Proof is over relative time, not absolute time
  - Absolute time can be achieved through online timestamping service

(Non-interactive) timestamping using VDFs

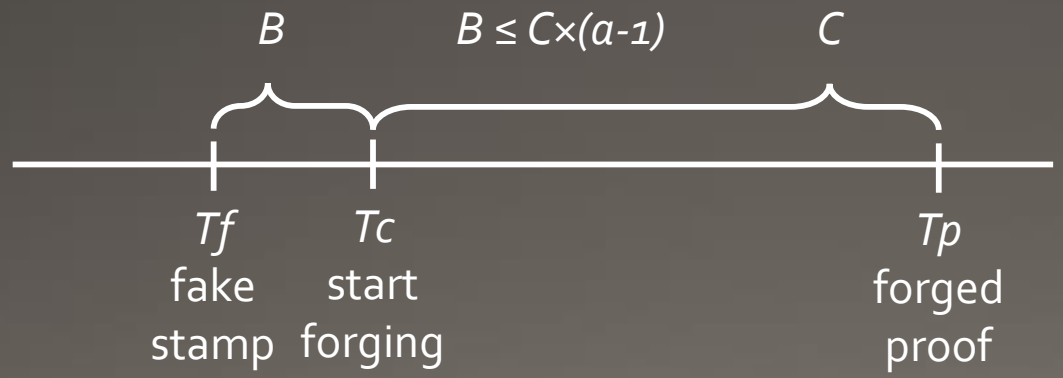
- [LS18,LSS20] Trusted Timestamping Service:
  - Hash-chain *ho*, *h1*, …
  - $h_i = Hash(\, h_{i-1} , M_i , time , VDF(h_{i-1} , T_i) , sig \,)$
  - Continuously compute VDF from last message until next message

  - Record a message *M* at time *Tr:*
    - Stop ongoing VDF and finalize VDF proof
    - Compute signature and next hash in chain
    - Continue VDF from new hash

  - Non-interactive timestamping proof for *M*:
    - Output hash-chain from *Tr* onwards
    - Copy ongoing VDF state, finalize and output VDF proof

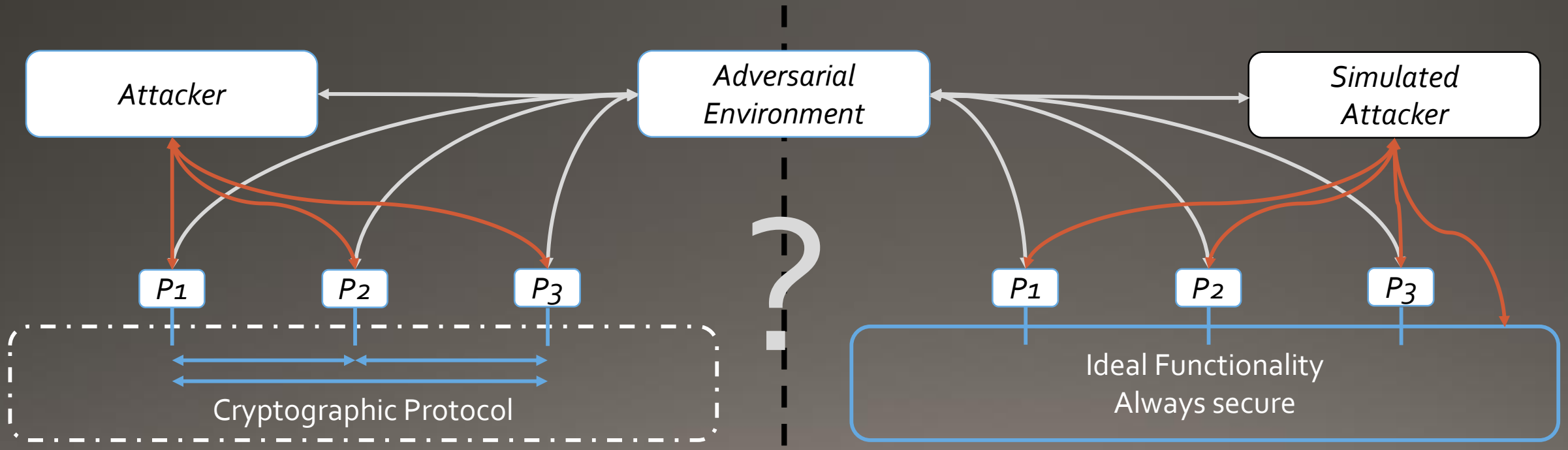
(Non-interactive) timestamping using VDFs

- [LSS20] Proven security bounding backdating attacks
  - Requires control signing key
  - Requires faster VDF core by factor $\alpha>1$
  - E.g. $\alpha=1.1$ then after 10 days at most backdated $10\times(\alpha-1) = 1$ day



(Non-interactive) timestamping using VDFs

- Proven secure in *Universal Composability Framework*
  - A Gold standard in Provable-Secure / Secure-by-Design protocols
  - Protocol is secure if no adversarial environment can distinguish between the real world (left) and the ideal world (right)



(Non-interactive) timestamping using VDFs

- Based on VDF => Only 1 VDF core sufficient
- Non-interactive proof based on hash-chains

  => Can be integrated into any other hash-chain protocol, such as DLT protocols at the block level

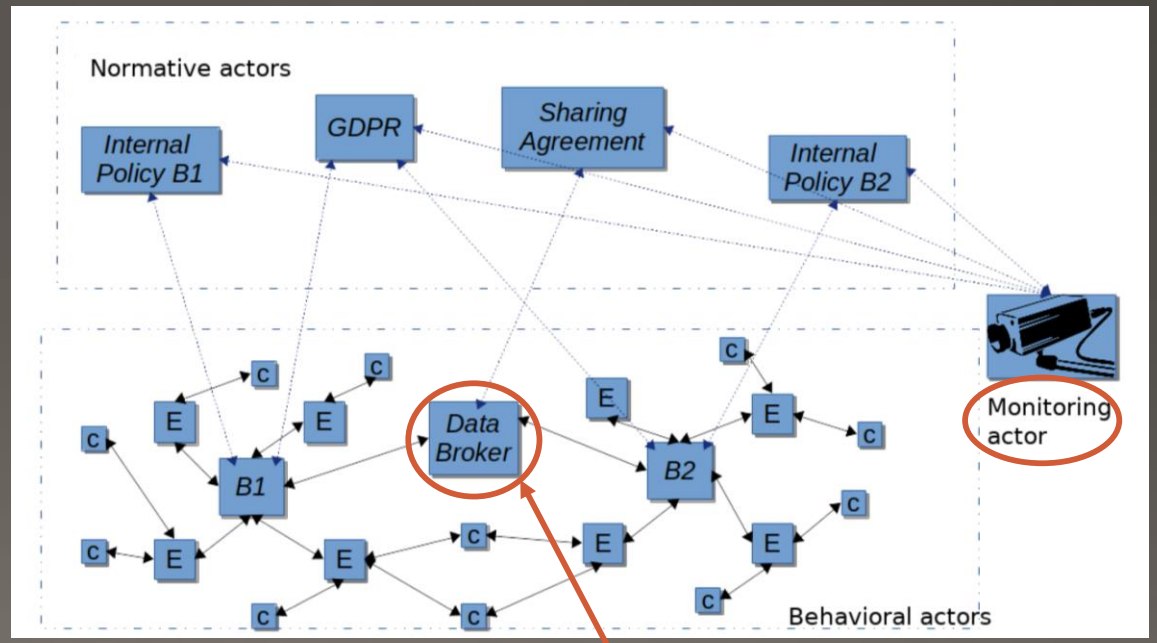
  => DLT immutability based on computational problem
  - Changing one block in DLT chain requires recomputing all subsequent VDF proofs

- [LS18] *On Immutability of Blockchains*
  - Shows protocol combining N timestamping servers
  - Adds threshold security: attackers needs to corrupt majority of *N* servers

WP1 Plans

WP1 Plans

- Activity 3 (ongoing)
  - Follow-up work on Timestamping
  - Prove secure VDF construction in Universal Composability Framework
  - Stronger foundations for underlying mathematical groups with hidden order
- Future activity 4
  - Extend eFLINT w/ *Trust Management*
  - Integration with WP3
- Future activity 5
  - Trusted third party "Data broker"
  - Replace with cryptographic protocol: *Secure Multi-Party Computation*
  - Integration with WP2



WP1 Plans

Activity 4: Identity management, authentication and authorization

- Goal: eFLINT extension Trust Management
  - Inspiration: KeyNote, Blaze et al., ’98

- DSL Extensions:
  - Principals (entities)
  - Credentials (pubkey, authorization, delegation)
  - Trust policy
  - Internal crypto processing

- Integration with WP3

WP1 Plans

Activity 5: Cryptographic mechanisms to share data privately

- eFLINT Demo uses Trusted Party for risk computation on multi-party data
- Goal: replace by *Secure Multi-Party Computation*
  - Generically possible but generally inefficient
  - Construct optimized protocol suitable for KYC usecase

- Strong guarantees:
  - Only output is learned
  - No other information is leaked
  - Unless > 1/3 of parties collude

- Integration with WP2: High-Performance Accelerators