A Secure and Private Proof-of-Location System

Mamunur Akand

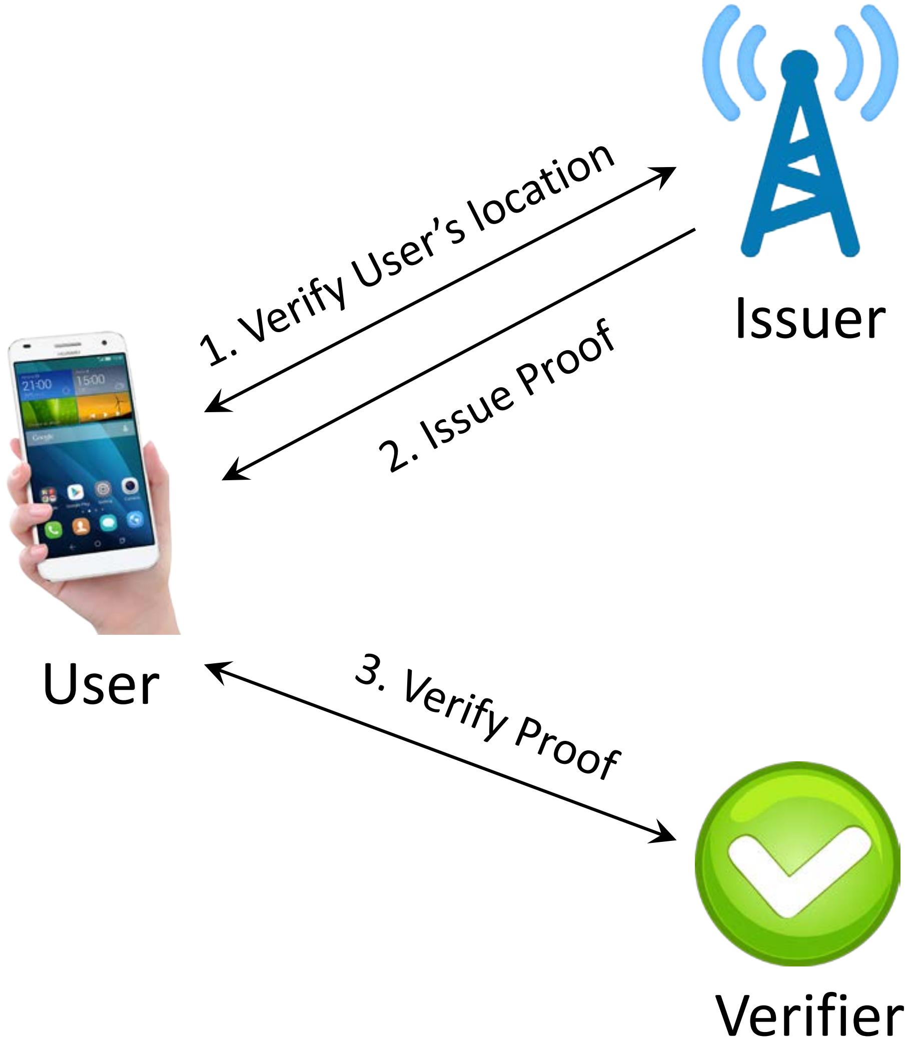
May 15, 2020

UNIVERSITY OF
CALGARY

Motivation

Proof-of-Location (*pol*)

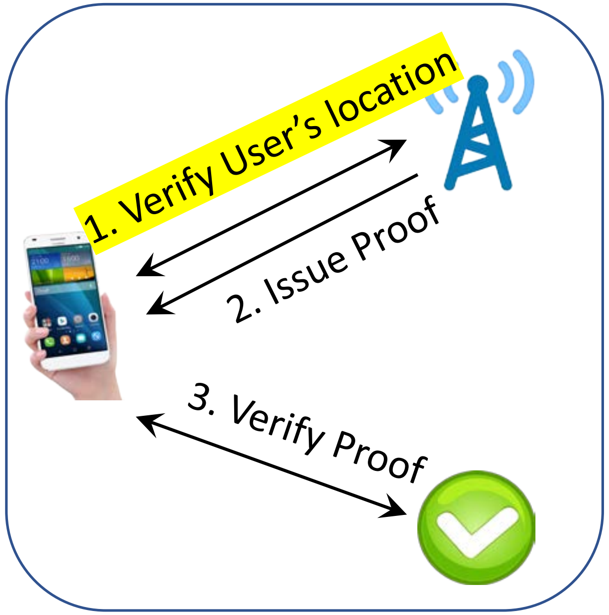
- Digital certificate, attesting location at a time
  - ➢ Reward system: discount to frequent customers
  - ➢ Supply chain: preserving product integrity
- Requirements:
  - ➢ Unforgeable
  - ➢ Non-transferable
  - ➢ User privacy

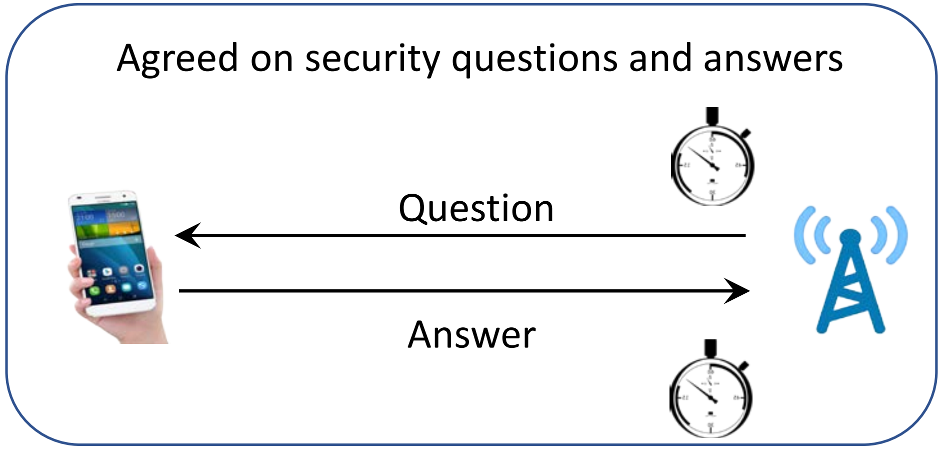


Background

Verify User's Location

- Device proximity based on network visibility [1, 2]
  - ➢ Can communicate ⇒ In proximity
  - ➢ Insecure: Relay attacks
- User-claimed GPS location [3]
  - Unreliable indoor
  - Known attacks on GPS
- Distance bounding protocols [4, 5]





Distance Bounding Protocol

[1] S. Sarioiu and A. Wolman. Enabling New Mobile Applications with Location Proofs. *HotMobile'09*.

[2] W. Luo and U. Hengartner. VeriPlace: A Privacy-aware Location Proof Architecture. *GIS'10*.

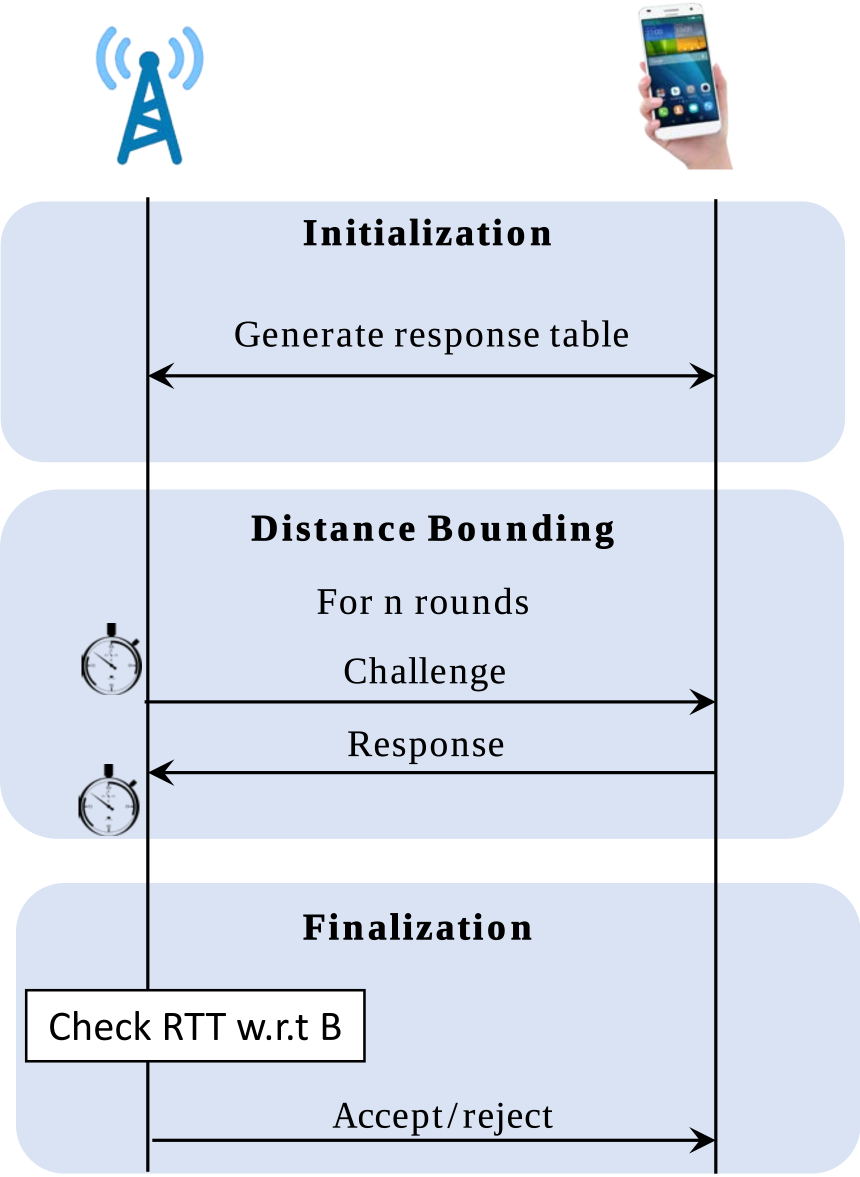
[3] Z. Zhu and G. Cao. APPLAUS: A Privacy-Preserving Location Proof Updating System for Location-Based Services. *INFOCOM'11*.

[4] X. Wang et al. STAMP: Ad hoc Spatial-Temporal Provenance Assurance for Mobile Users. *ICNP'13*.

[5] S. Gambs et al. PROPS: A PRivacy-Preserving Location Proof System. *SRDS'14*.

Background

Distance-Bounding Protocol



Initialization

Generate response table

Distance Bounding

For n rounds

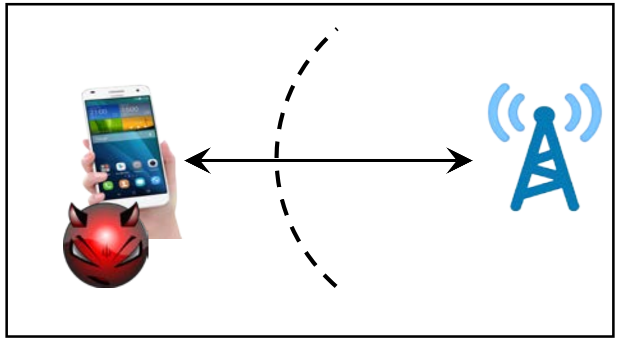
Challenge

Response

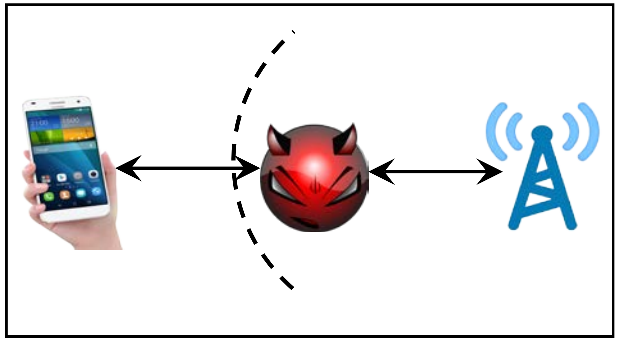
Finalization

Check RTT w.r.t B

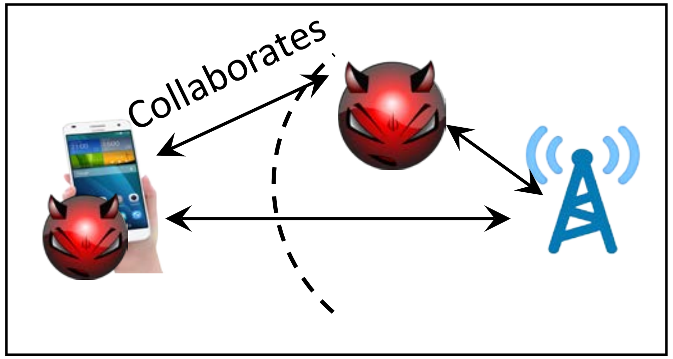
Accept/reject



Distance Fraud



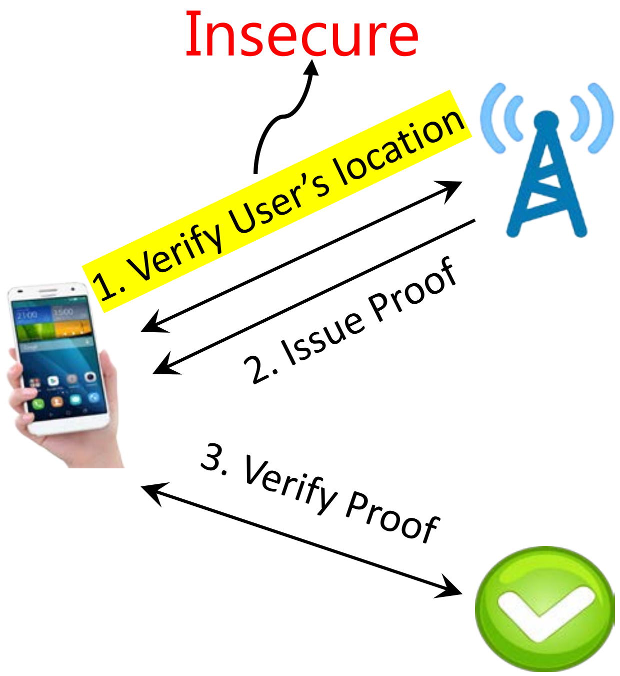
Mafia Fraud



Terrorist Fraud

Shortcomings of Existing POL Systems

- User's location verification not secure [1, 2, 3].
- Systems in [4,5] use DB from [6].
  - ➢ Insecure against Distance fraud, Terrorist Fraud [7]
  - ➢ Cannot replace with secure DB
- No common model for security and privacy
  - ➢ Informally specified properties
  - ➢ Different terms for same property



[1] S. Sarioiu and A. Wolman. Enabling New Mobile Applications with Location Proofs. *HotMobile'09*.
[2] W. Luo and U. Hengartner. VeriPlace: A Privacy-aware Location Proof Architecture. *GIS'10*.
[3] Z. Zhu and G. Cao. APPLAUS: A Privacy-Preserving Location Proof Updating System for Location-Based Services. *INFOCOM'11*.
[4] X. Wang et al. STAMP: Ad hoc Spatial-Temporal Provenance Assurance for Mobile Users. *ICNP'13*.
[5] S. Gambs et al. PROPS: A PRivacy-Preserving Location Proof System. *SRDS'14*.
[6] L. Bassard and W. Bagga. Distance-Bounding Proof of Knowledge to Avoid Real-time Attacks. *IFIP'05*.
[7] A. Bay et al. The Bussard-Bagga and Other Distance-Bounding Protocols Under Attacks. *ICISC'12*.

Our Contribution

Formalize security and privacy of POL systems.

Construct a POL that provably achieves these properties.

Implement cryptographic algorithms to show feasibility of the solution.

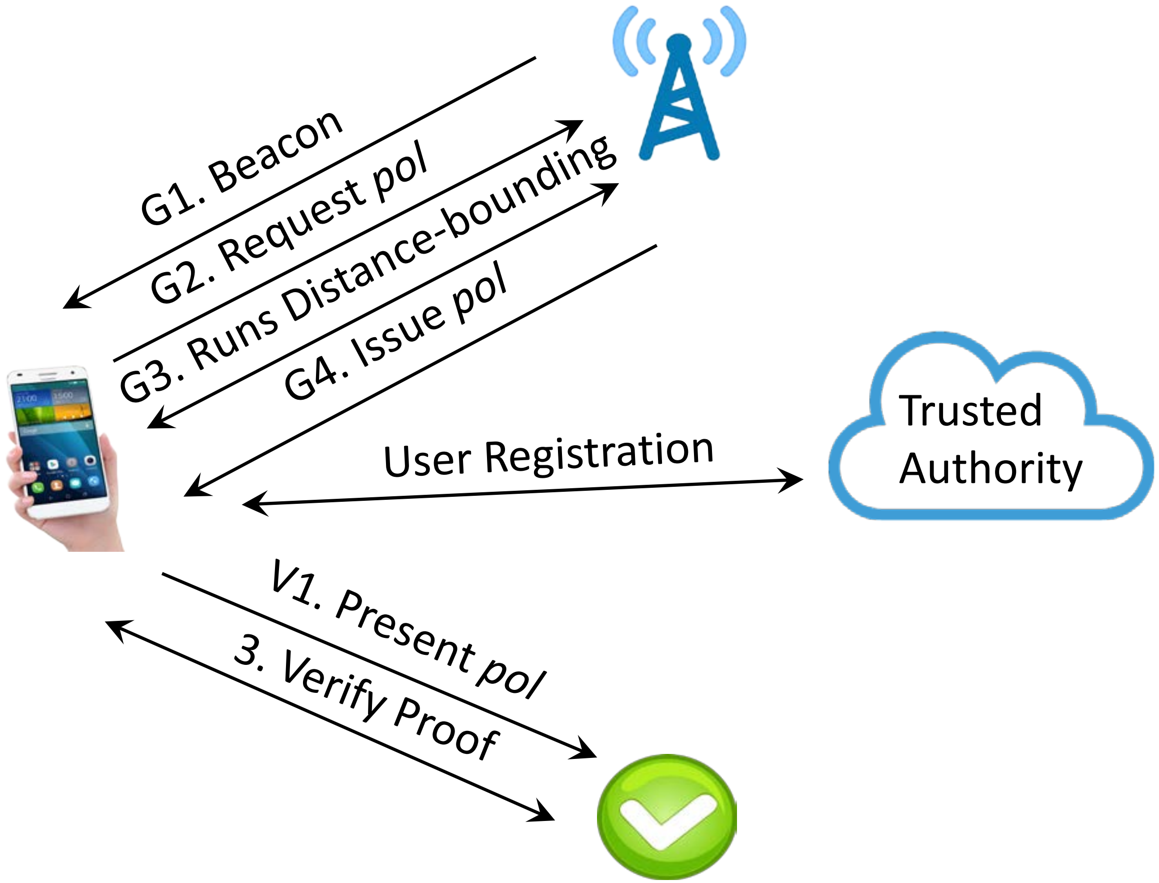
System Model

Entities:

- Trusted authority
  - System parameters
  - Keys, certificates for entities
- User
- Issuer
  - Access point (AP)
- Verifier
  - Service provider

Trust Assumption:

- Issuer, verifier: honest, curious
- User: untrusted



Other Assumptions:

- User u's location is w.r.t the location of issuing Access Point ap
- *pol* is ap's signature on *"u is within distance B from* loc_{ap}*"*

Definitions

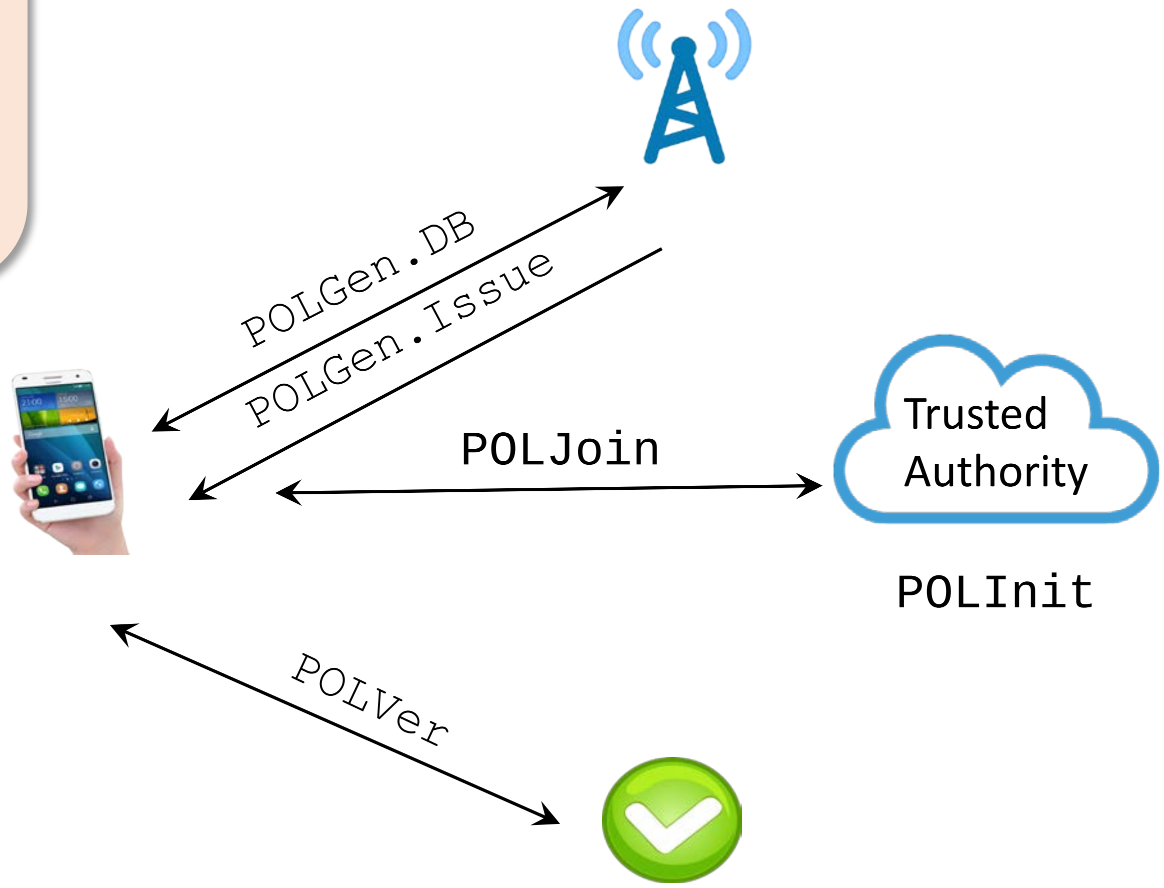
Definition 1 (POL Scheme). *Defined by*

$\texttt{POLInit}(1^\lambda) \rightarrow$ *public and private parameters*

$\texttt{POLJoin}[TA \leftrightarrow User]$: *User registration*

$\texttt{POLGen}[User \leftrightarrow Issuer]$: $\texttt{POLGen.DB}$, $\texttt{POLGen.Issue}$

$\texttt{POLVer}\ [User \leftrightarrow Verifier]$: *proof verification*

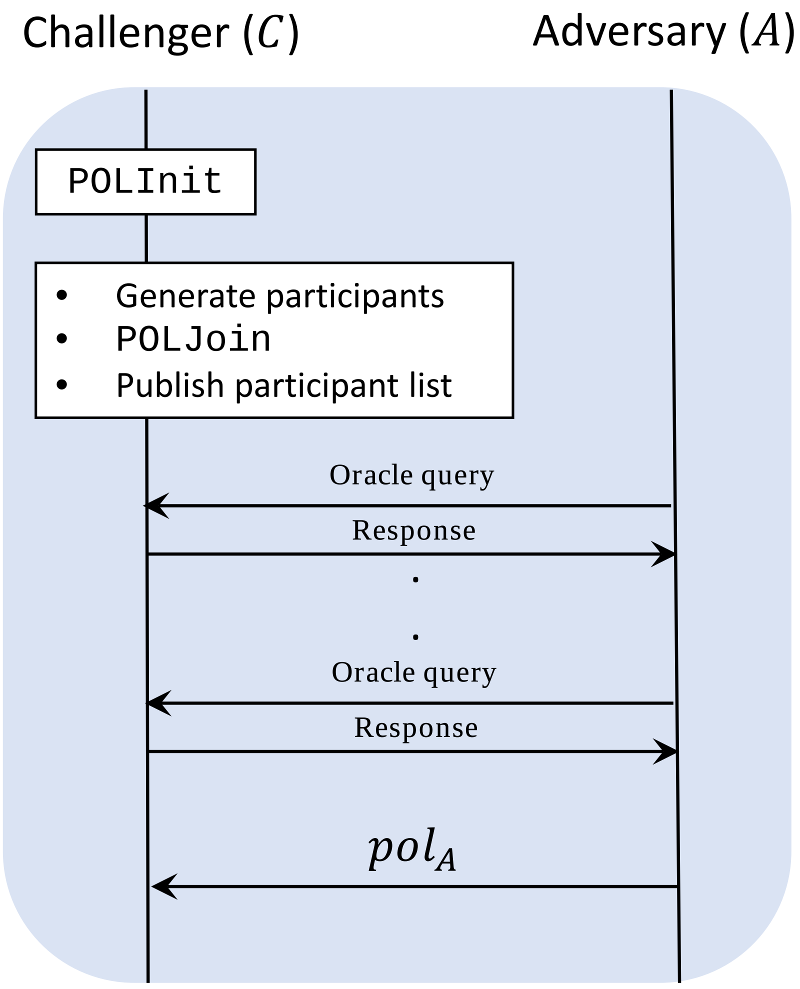


Definitions

Oracle query	Output	Update List
Corrupt(X)	Credentials of X	CorruptList$\langle X\rangle$
POLGen(ap, u)	$pol \leftarrow \texttt{POLGen}[u \leftrightarrow ap]$	GenList$\langle pol, u\rangle$
POLVer (u, v, pol)	$pol \leftarrow \texttt{POLVer}[u \leftrightarrow ap]$	VerList$\langle pol, u\rangle$

Definition 2 (POL Game). *Define a challenger-adversary game as:*

1. *Initialize*
   - *Challenger runs* `POLInit`
2. *Generate participants*
   - *Challenger generates users, issuers and verifiers*
   - *Challenger runs* `POLJoin` *for all users*
3. *Queries*
   - *Adversary makes oracle queries*
4. *Adversary outputs*
   - *Adversary outputs a proof-of-location* pol_A

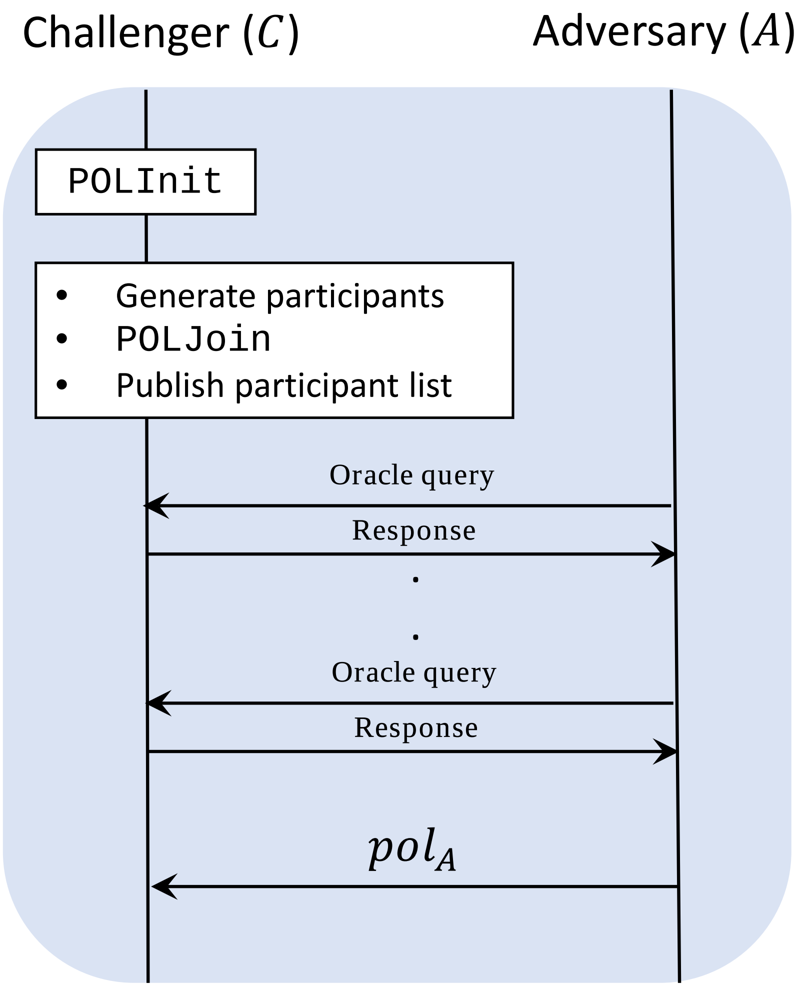


POL Game

POL Properties

POL Security properties

- Unforgeability, Non-transferability, Anonymity
- Defined based on the game
- Indistinguishability based approach for user anonymity
  - w.r.t to issuer
  - w.r.t to verifier



POL Game

POL Properties

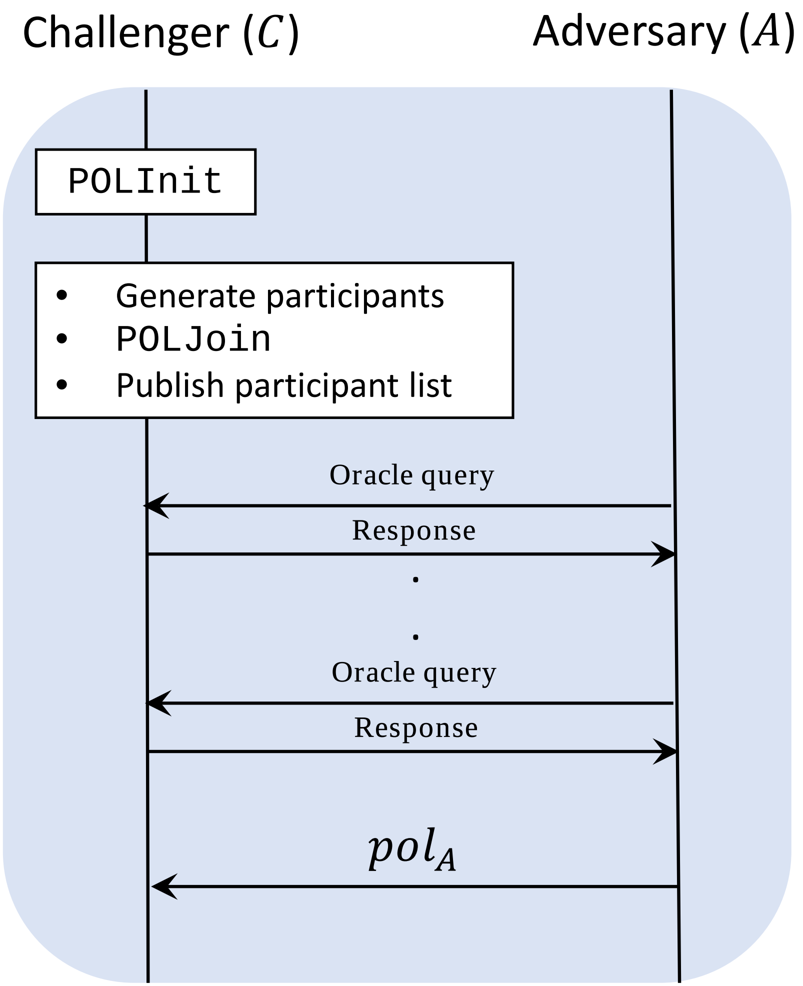
Property 1 (POL Unforgeability). Consider a POL scheme and a POL game where

- Corrupt(X) query only corrupts users
- Adversary outputs pol_A.
- Winning conditions:
  - $\exists\ (pol, .) \in VerList\ s.t.\ pol = pol_A$
  - $\nexists (pol, .) \in GenList\ s.t.\ pol = pol_A$ OR $\exists\ (pol, .) \in GenList\ s.t.\ pol = pol_A \wedge d(u, ap) > B$

POL is Unforgeable if adversary wins with negligible probability.

Adversary wins if:

- pol_A is successfully verified
- pol_A is not generated by a listed issuer, Or,
- pol_A is generated by a listed issuer, but user was far away from issuer



- Generate participants
- POLJoin
- Publish participant list

POL Game

POL Properties

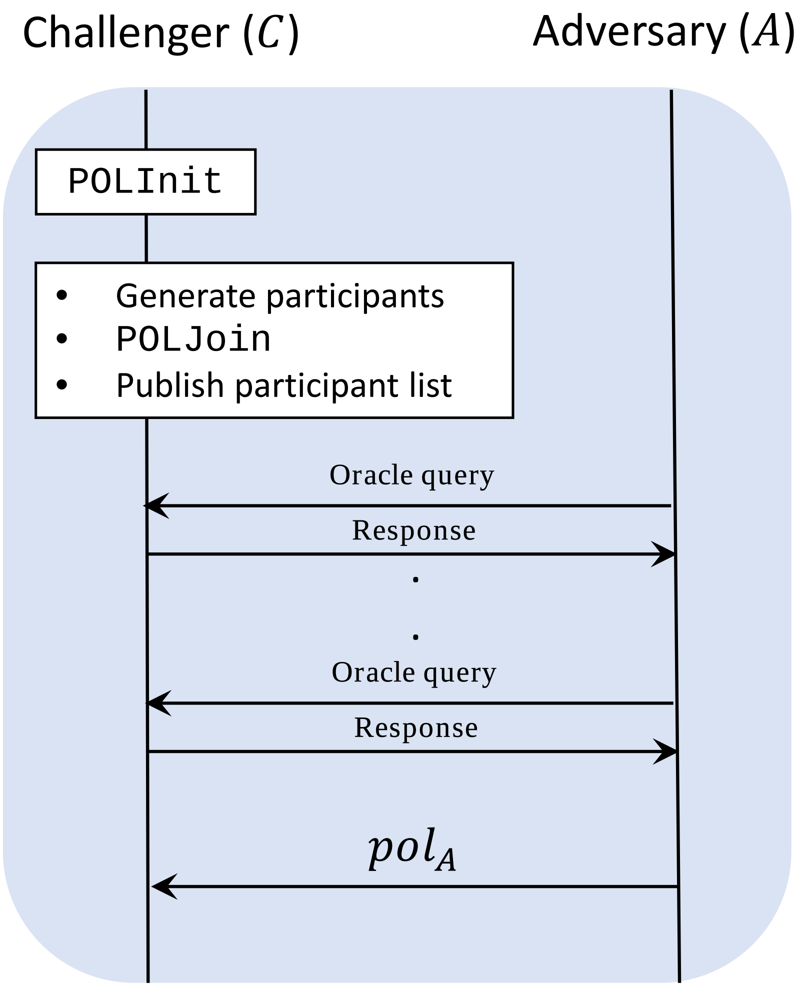
Property 2 (POL Non-transferability). Consider a POL scheme and a POL game where

- Corrupt(X) query only corrupts users
- Adversary outputs pol_A.
- Winning conditions:
  - $\exists\ (pol, u) \in VerList\ s.t.\ pol = pol_A$
  - $\exists (pol, u') \in GenList\ s.t.\ pol = pol_A \wedge u' \neq u$

POL is Non-transferable if adversary wins with negligible probability.

Adversary wins if:

- pol_A is successfully verified for user u
- pol_A was issued to user $u' \neq u$



POL Game

POL Properties

Property 1 (POL Anonymity). Consider a POL scheme and a POL game where

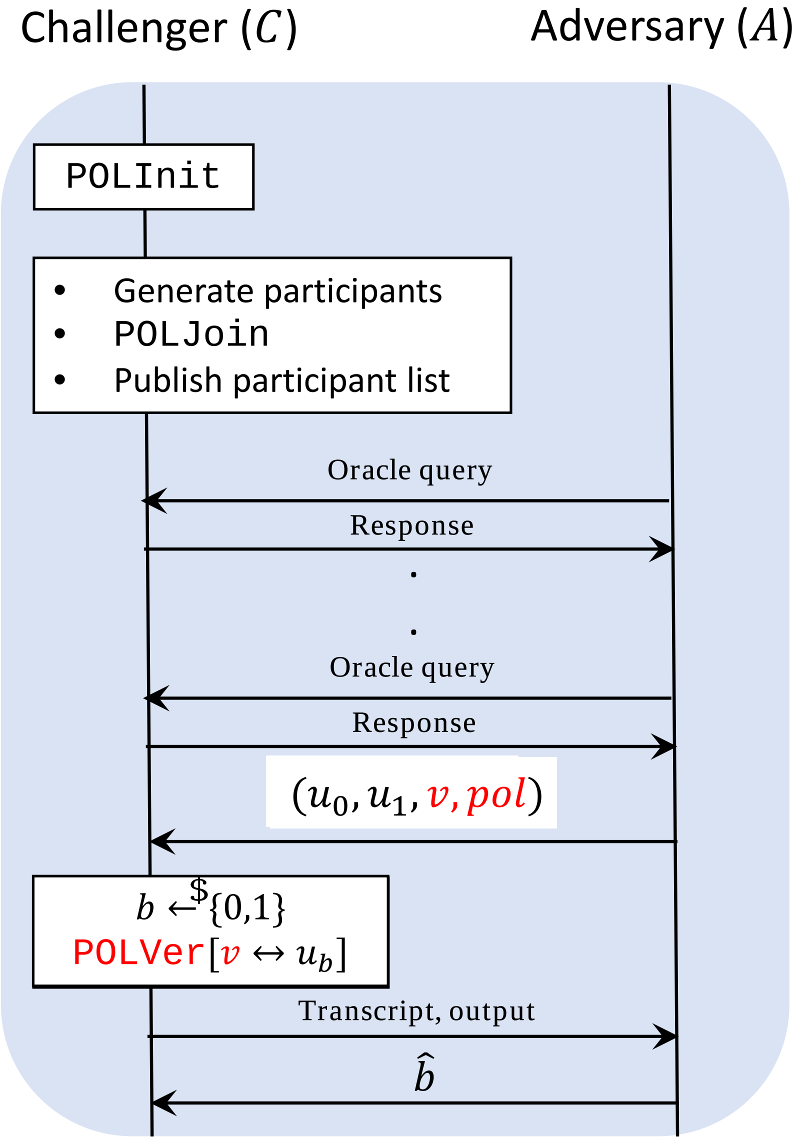
- Corrupt(X) query only corrupts issuers and verifiers

Anonymity w.r.t verifier:

- Adversary chooses a pair of users (u_0, u_1) and a verifier v
- Challenger runs POLVer between v and $u_{b \leftarrow \{0,1\}}$ for pol
- Transcript and output of protocol are returned to A.
- Adversary outputs $\hat{b}$

Winning condition:

- $|\Pr[\hat{b} = b] - \frac{1}{2}|$ is non-negligible.



POL Game

POL Construction

Cryptographic primitives

- Digital signature (KeyGen, Sign, Verify) [8]
- Commitment (KeyGen, Commit) [9]
  - Committer hides a value **x (com = Commit(x,r))**
  - Reveal **x** later
  - No info on **x** is leaked before reveal stage (hiding)
  - **x** cannot be changed once it is committed (binding)
- Zero-knowledge proof of knowledge
  - Prover-verifier protocol
  - Prover possess w that satisfies relation R
  - No info on w revealed
  - $ZKPoK\{(\alpha,\beta,\gamma): y = g^{\alpha}h^{\beta} \wedge \tilde{y} = \tilde{g}^{\alpha}\tilde{h}^{\gamma}\}$

[8] J. Camenisch et al. A signature scheme with efficient protocols. *SCN'02*.

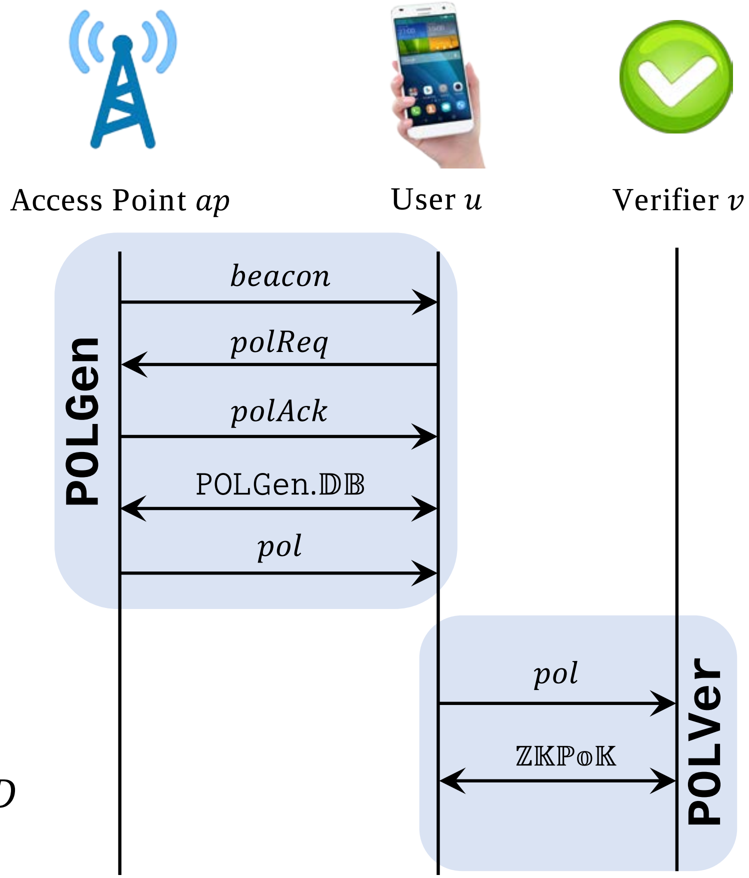
[9] E. Fujisaki et al. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. *EUROCRYPT'98*

POL Construction

1. $\mathtt{POLInit}(1^{\lambda})$
   - *TA Generates its public/private signature keypair* (pk^{TA}, sk^{TA})
2. $\mathtt{POLJoin}[TA \leftrightarrow User]$
   - *TA issues secret* s_u *and certificate* $cert_u$ *to the user* u
3. $\mathtt{POLGen}[User \leftrightarrow Issuer]$
   - `POLGen.DB`
   - `POLGen.Issue`
4. $\mathtt{POLVer}\ [User \leftrightarrow Verifier]$

$$\mathbb{ZKPoK}\{(s_u, \alpha, cert_u) : com = g^{s_u} h^{\alpha} \wedge \mathbb{DS}.\mathrm{Vf}(pk^{TA}, s_u, cert_u) = 1\}$$ [8]

$beacon \coloneqq seqID|pk^{ap}$
$polReq \coloneqq req|N_u|SeqID$
$polAck \coloneqq ack|N_{ap_0}|seqID$
$pol \coloneqq sig|msg,$
$sig \coloneqq \mathbb{DS}.\mathrm{Sig}\,(sk^{ap}, msg)$
$msg \coloneqq com|pk^{ap}|\ell oc_{ap}|t|N_{ap}$



POLGen and POLVer

POL Construction

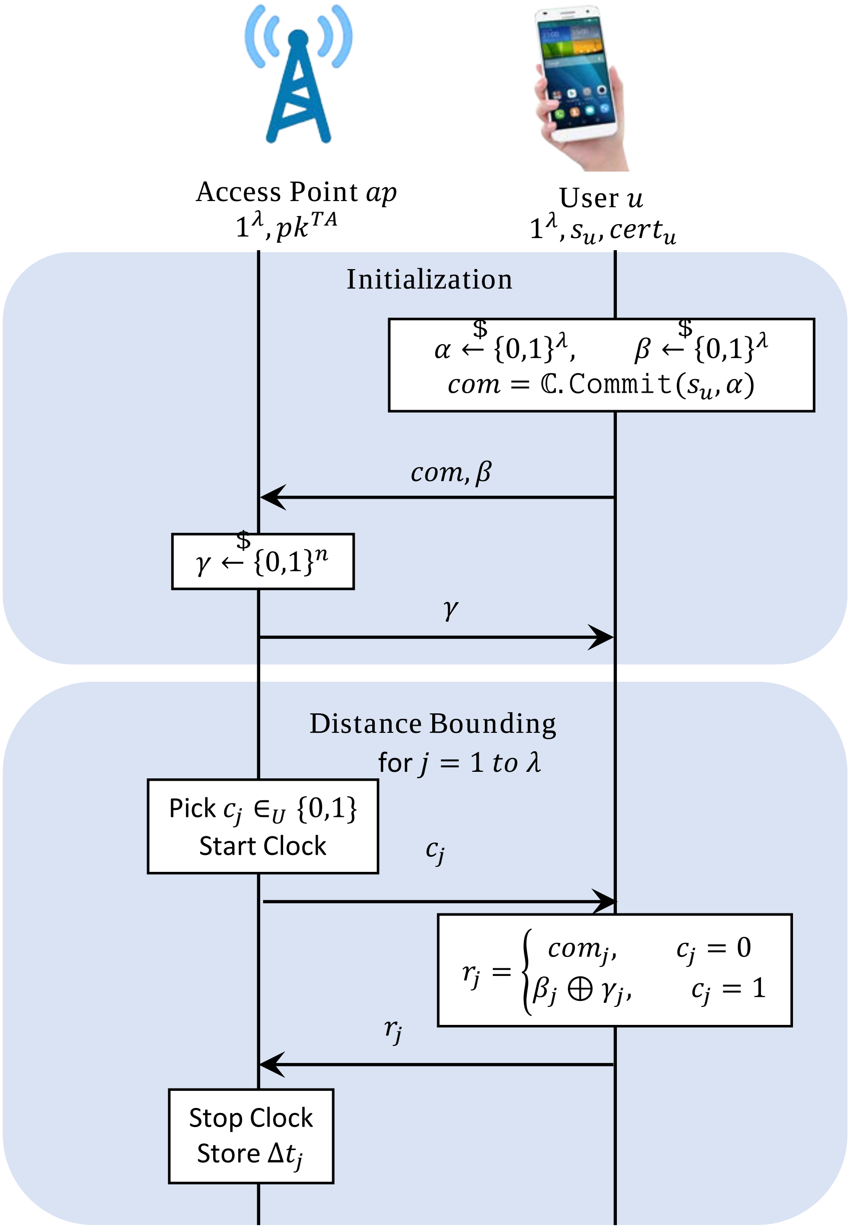
Requirements:

- Distance bounding protocol:
  1. User anonymity
  2. Transcript with sufficient information ⇒ make *pol* non-transferable
- Cannot use existing anonymous DB [9,10,11]
  - Do not satisfy both properties

[9] Ahmadi, A., et al.: New attacks and secure design for anonymous distance-bounding. ACISP'18
[10] Bultel, X., et al.: A prover-anonymous and terrorist-fraud resistant distance-bounding protocol. ACM WiSec'16
[11] Gambs, S., et al.: Prover anonymous and deniable distance-bounding authentication. ASIACCS'14

POL Construction

POLGen.DB



Access Point ap
$1^{\lambda}, pk^{TA}$

User u
$1^{\lambda}, s_u, cert_u$

Initialization

$\alpha \xleftarrow{\$} \{0,1\}^{\lambda}, \quad \beta \xleftarrow{\$} \{0,1\}^{\lambda}$
$com = \mathbb{C}.\mathtt{Commit}(s_u, \alpha)$

com, β

$\gamma \xleftarrow{\$} \{0,1\}^{n}$

γ

Distance Bounding
for $j = 1$ *to* λ

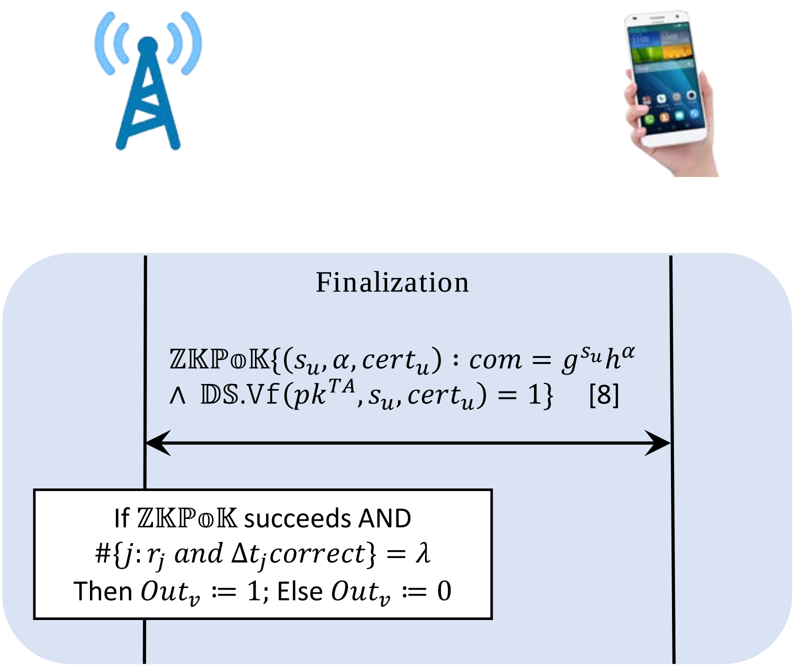
Pick $c_j \in_U \{0,1\}$
Start Clock

c_j

$r_j = \begin{cases} com_j, & c_j = 0 \\ \beta_j \oplus \gamma_j, & c_j = 1 \end{cases}$

r_j

Stop Clock
Store Δt_j



Finalization

$\mathbb{ZKPoK}\{(s_u, \alpha, cert_u) : com = g^{s_u} h^{\alpha}$
$\wedge\ \mathbb{DS}.\mathtt{Vf}(pk^{TA}, s_u, cert_u) = 1\}$ [8]

If $\mathbb{ZKPoK}$ succeeds AND
$\#\{j : r_j \ and \ \Delta t_j correct\} = \lambda$
Then $Out_v \coloneqq 1$; Else $Out_v \coloneqq 0$

ZKPoK: com is a valid commitment over a value s_u and s_u is certified by the TA

[8] J. Camenisch et al. A signature scheme with efficient protocols. *SCN'02*.

Security Analysis

Theorem.

i. *DB Security:* `POLGen.DB` *is secure against distance fraud, mafia fraud and terrorist fraud attacks.*

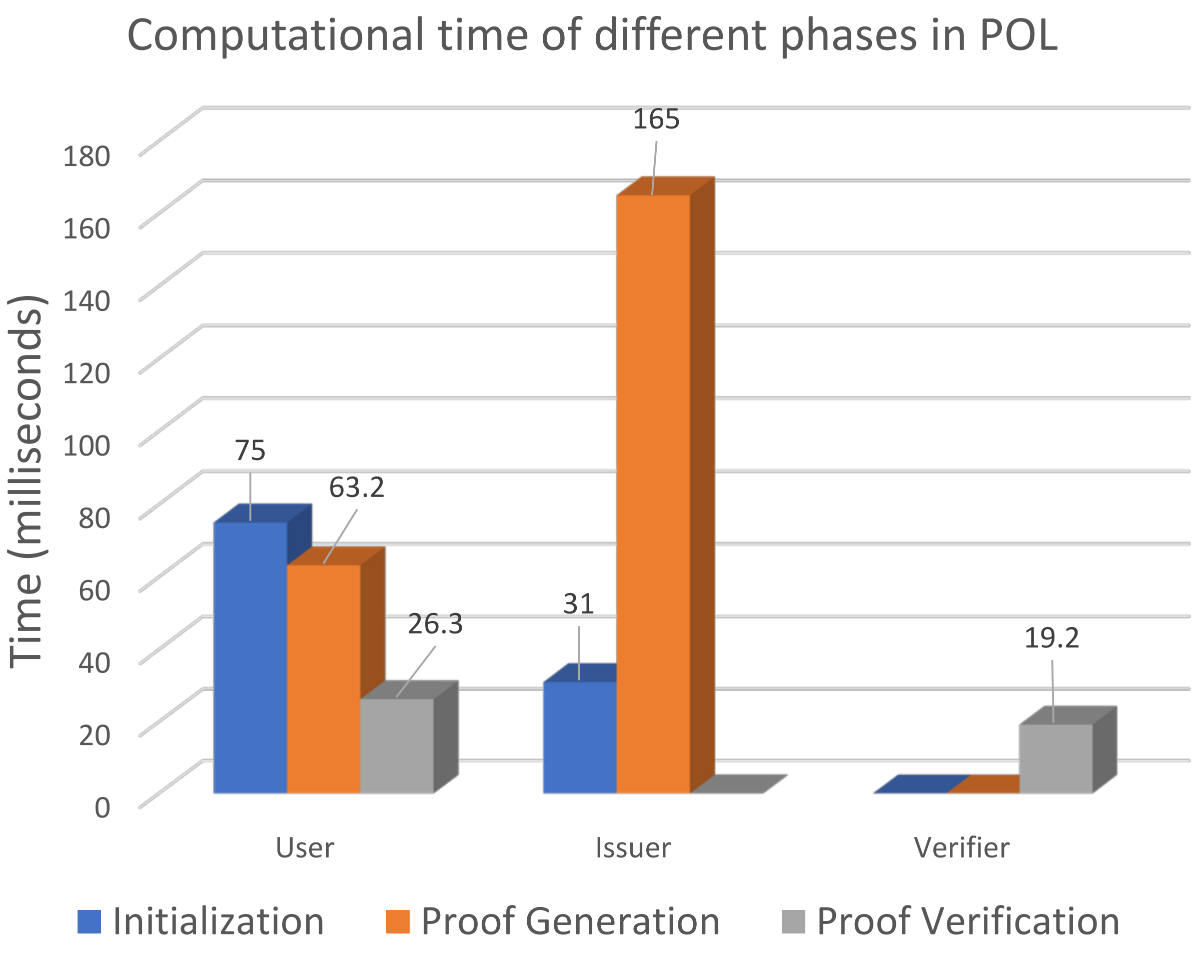
ii. *POL Unforgeability: Assuming* `POLGen.DB` *is secure and digital signature is secure, POL is unforgeable.*

iii. *POL Non-transferability: Assuming the* ZKPoK *is sound, and user does not share credential, POL is Non-transferable.*

iv. *POL Anonymity: Assuming the commitment scheme is computationally hiding and ZKPoK is zero knowledge, POL is anonymous w.r.t issuer and verifier.*

Proof-of-concept Implementation

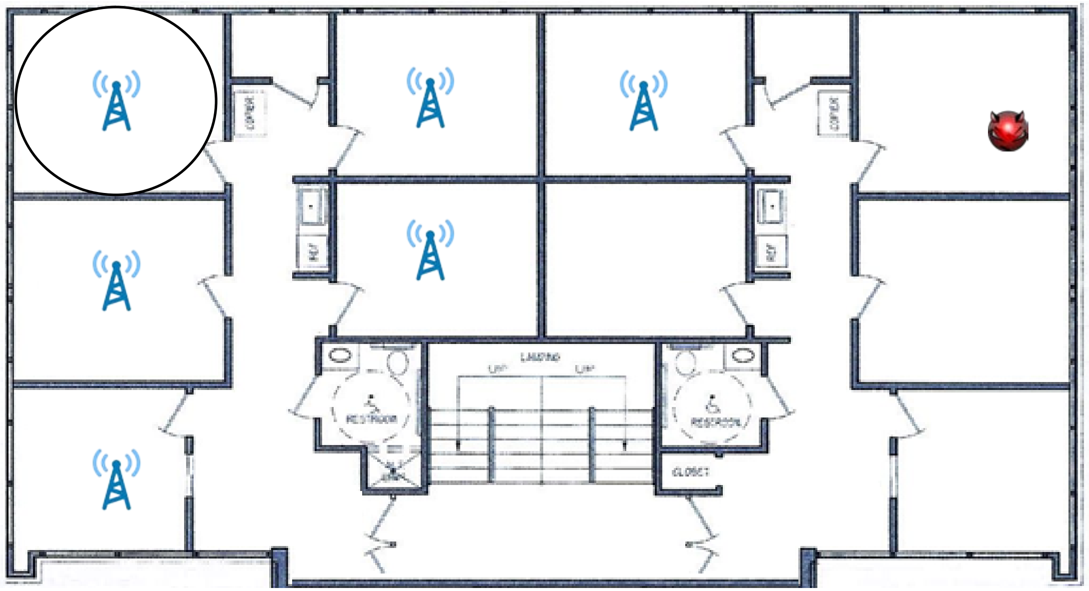
- Idemix Java Library (www.zurich.ibm.com/idemix)
  - ➢ Commitment
  - ➢ ZKP
  - ➢ CL-signatures
- Samsung Galaxy S9
- No DB
  - ➢ Device proximity based on network visibility
- Initialization:
  - ➢ Commitment, ZKPoK
- Proof Generation:
  - ➢ CL-signature
- Proof verification:
  - ➢ ZKPoK
- Proof size: 1940 bytes



Other Contributions

Geo-tampering attack

- Physically move issuer ⇒ forge proof-of-location



- Solution: Ensure that issuer’s relative position to its neighbors is unchanged