Group Time-based One-time Passwords and its Application to Efficient Privacy-Preserving Proof of Location

Zheng Yang[1], **Chenglu Jin**[2], Jianting Ning[3]

Zengpeng Li[4], Dinh Tien Tuan Anh[5], Jianying Zhou[5]

[1]Southwest University, China

[2]CWI Amsterdam, Netherlands

[3]Fujian Normal University, China

[4]Shandong University, China

[5]Singapore University of Technology and Design, Singapore

Overview

- **Background**

Overview

- **Background**
- **Group Time-based One-Time Passwords (GTOTP)**

Overview

- **Background**
- **Group Time-based One-Time Passwords (GTOTP)**
- **Privacy-Preserving Proof of Location**

Overview

- **Background**
- **Group Time-based One-Time Passwords (GTOTP)**
- **Privacy-Preserving Proof of Location**
- **Evaluation**

Overview

- **Background**
- **Group Time-based One-Time Passwords (GTOTP)**
- **Privacy-Preserving Proof of Location**
- **Evaluation**
- **Summary and Open Questions**

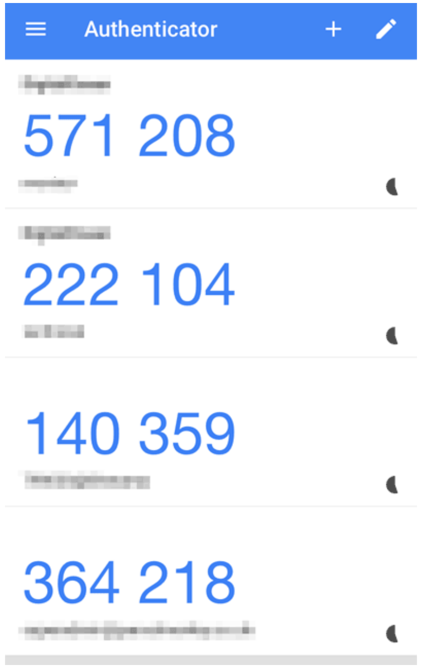
Time-based One-time Passwords (TOTP)

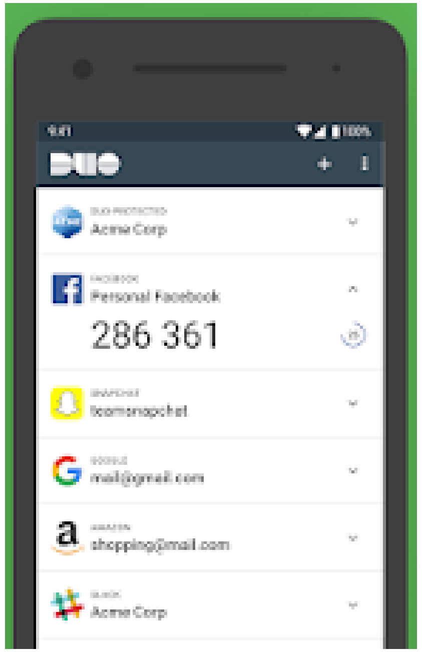
Time-based One-time Passwords (TOTP)

- Time-based One-time Passwords

Time-based One-time Passwords (TOTP)

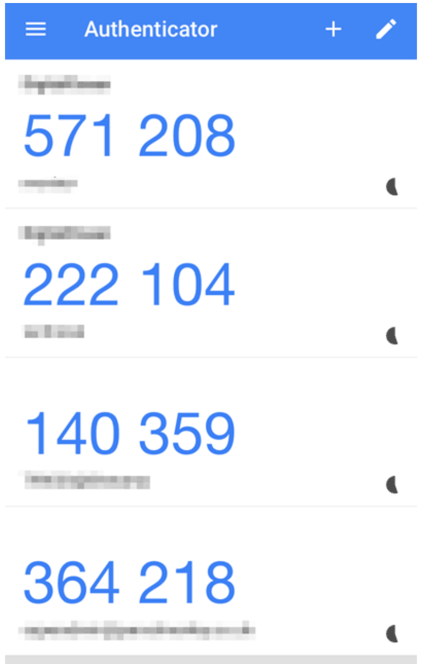
- Time-based One-time Passwords
- TOTP as an authentication factor:
  - Lightweight: very efficient to generate
  - Easy to use

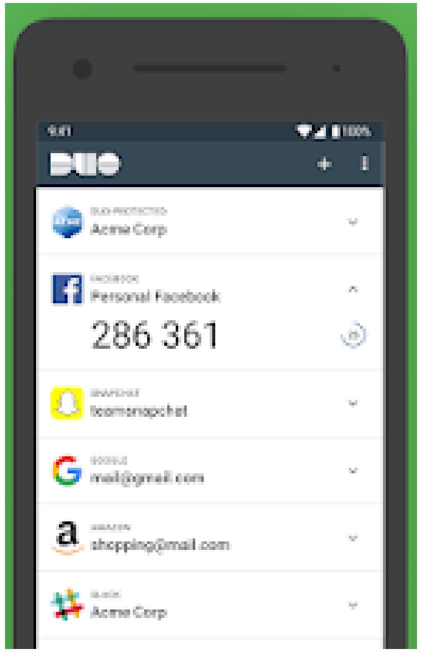




Time-based One-time Passwords (TOTP)

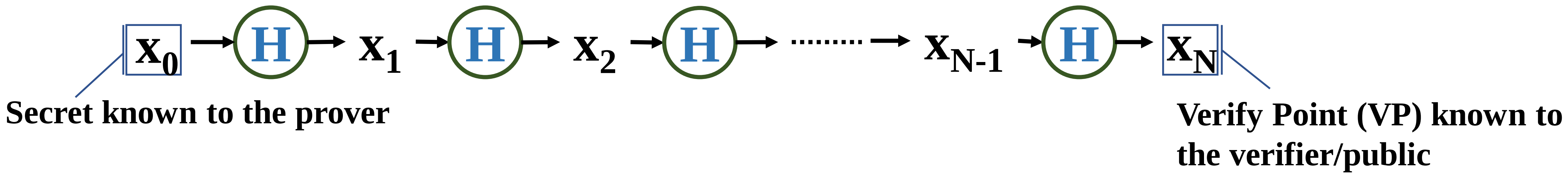
- Time-based One-time Passwords
- TOTP as an authentication factor:
  - Lightweight: very efficient to generate
  - Easy to use
- TOTP can be realized using
  - Symmetric keys shared between the prover and the verifier
  - Asymmetric method: **hash-based** or digital signatures





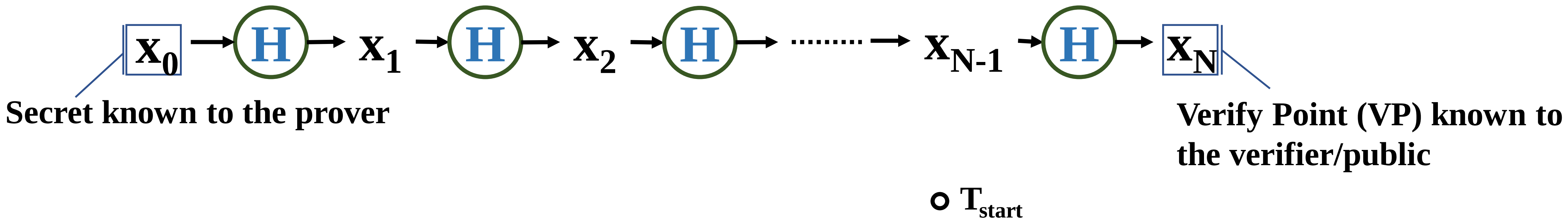
Time-based One-time Passwords

- Traditional hash-based TOTPs



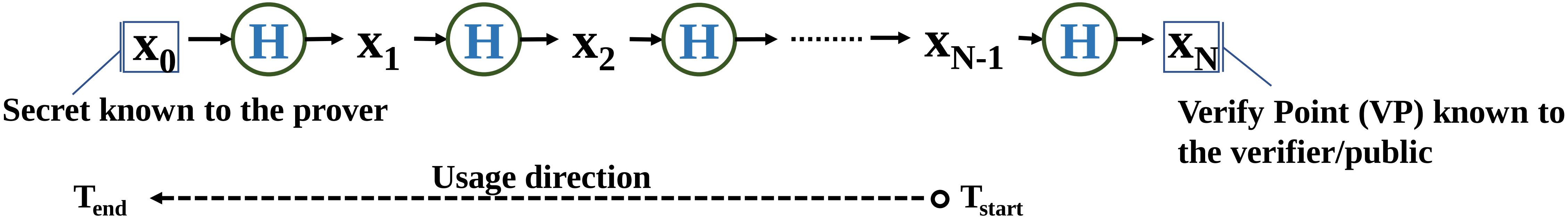
Time-based One-time Passwords

- Traditional hash-based TOTPs



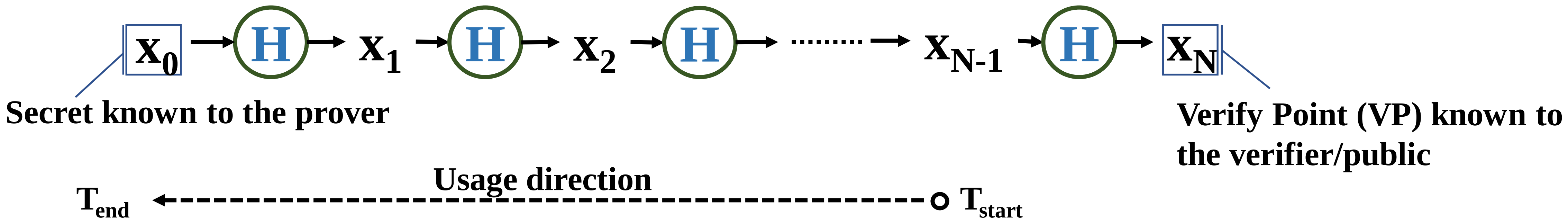
Time-based One-time Passwords

- Traditional hash-based TOTPs



Time-based One-time Passwords

- Traditional hash-based TOTPs



- One key pair per user (x_0, x_N)
  - Asymmetric: verifier compromise resilience
  - **No identity privacy:** each ***verify point*** x_N is associated with one prover, and the verifier knows the identity of the prover

TOTP with Privacy?

TOTP with Privacy?

- Group Signature: privacy-preserving signatures
  - Computationally expensive: many exponentiations or pairings
  - Not fit for resource-constrained devices or applications

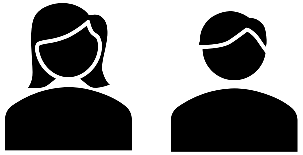
TOTP with Privacy?

- Group Signature: privacy-preserving signatures
  - Computationally expensive: many exponentiations or pairings
  - Not fit for resource-constrained devices or applications

How to *efficiently* and *generically* transform a traditional (asymmetric) TOTP into a GTOTP scheme?

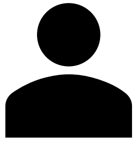
Group TOTP

Group Members (Provers)



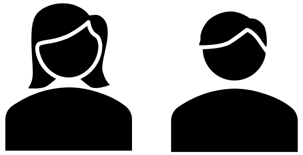
Trusted Registration Authority (RA)

Verifier



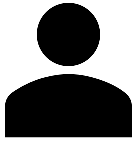
Group TOTP

Group Members (Provers)



Trusted Registration Authority (RA)

Verifier

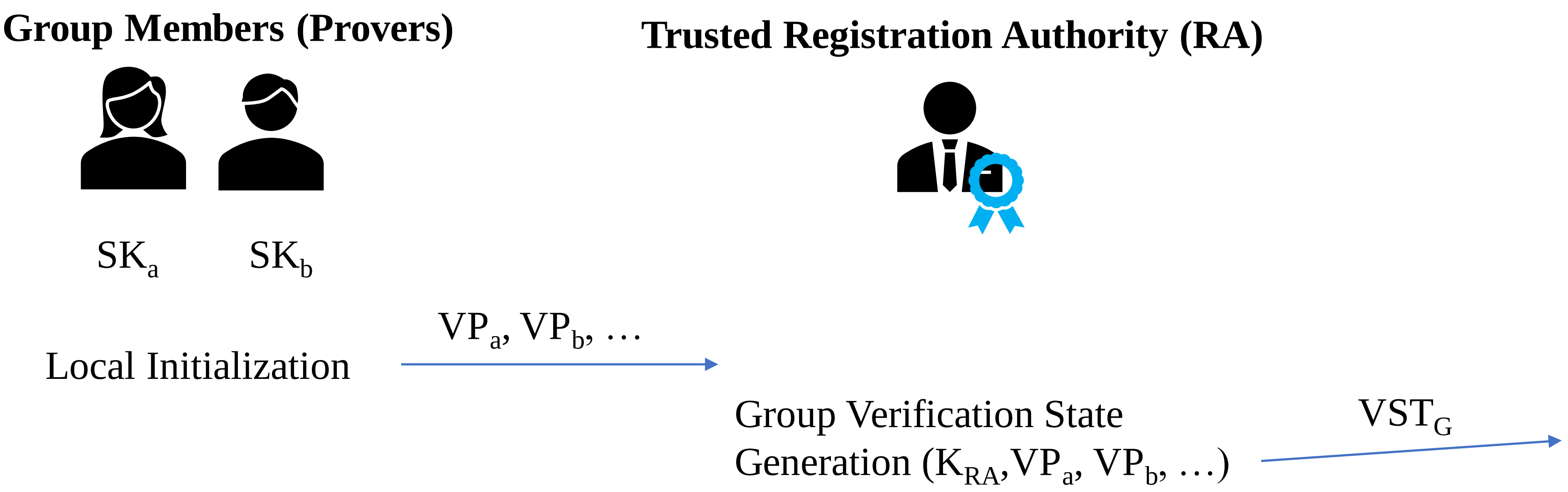


Local Initialization → VP_a, VP_b, …

Group TOTP

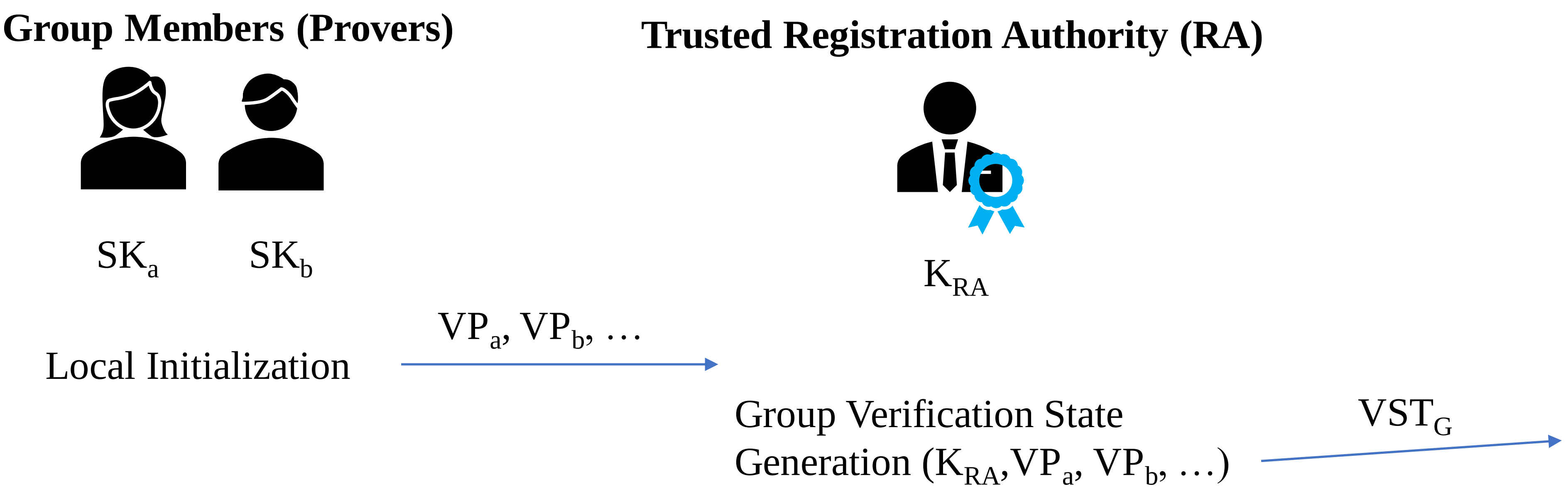
Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK$_a$ SK$_b$
VP$_a$, VP$_b$, …
Local Initialization

Group TOTP

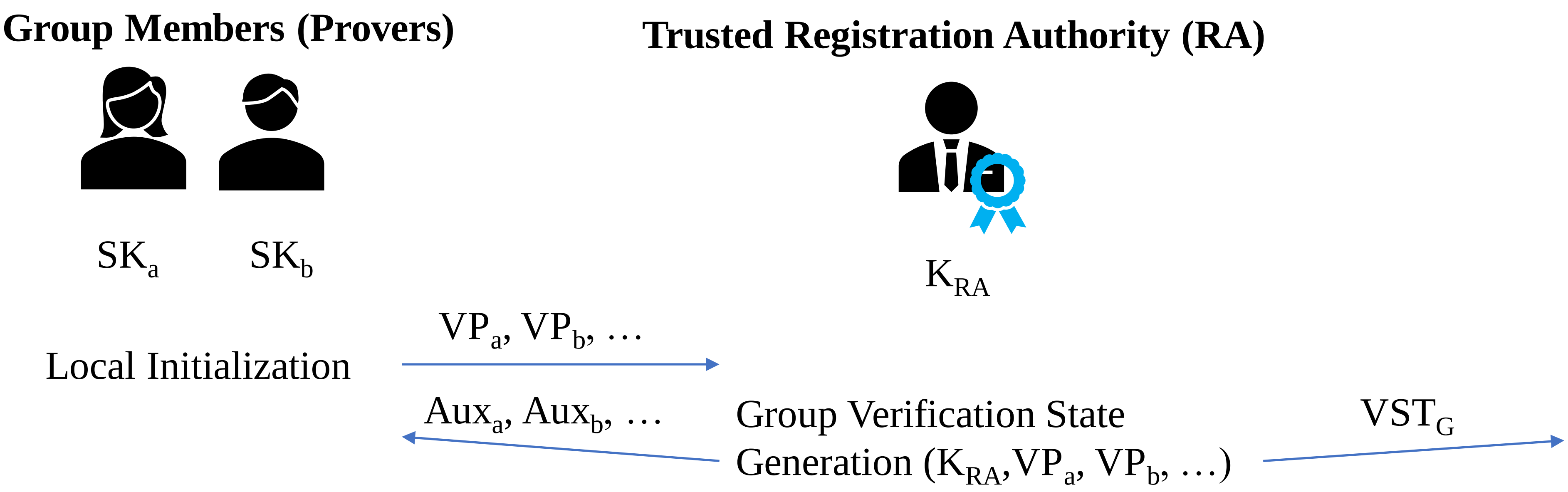


Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
SK_b
Local Initialization
VP_a, VP_b, …
Group Verification State Generation (K_{RA},VP_a, VP_b, …)
VST_G

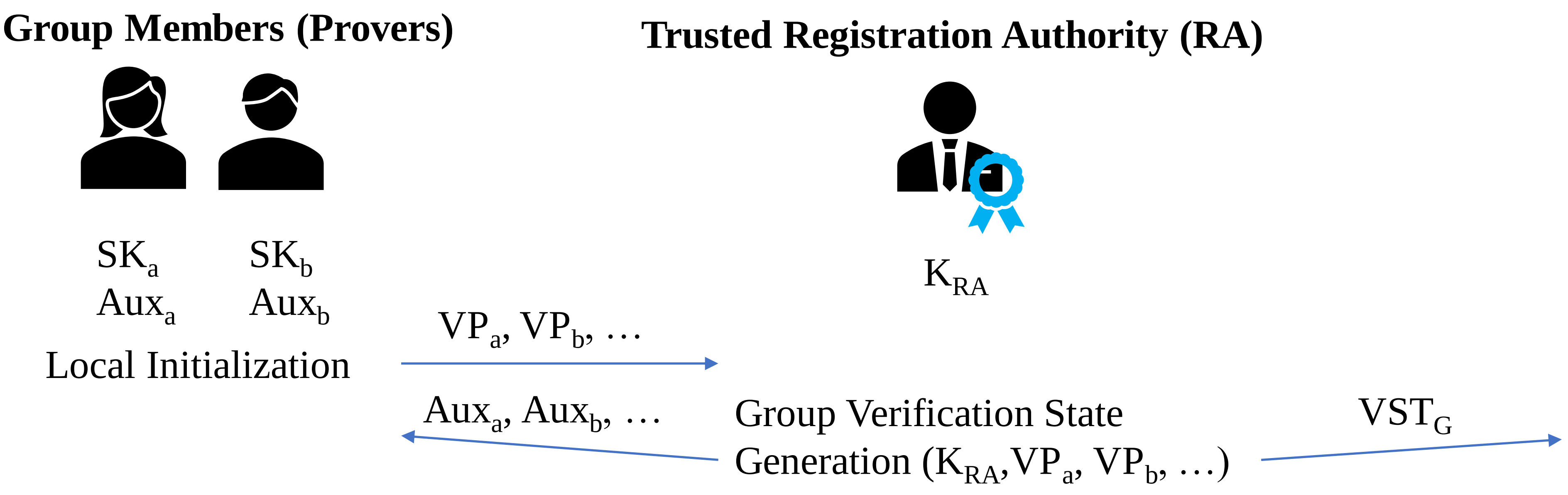
Group TOTP


Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
SK_b
K_{RA}
VST_G
VP_a, VP_b, …
Local Initialization
Group Verification State
Generation (K_{RA},VP_a, VP_b, …)
VST_G

Group TOTP

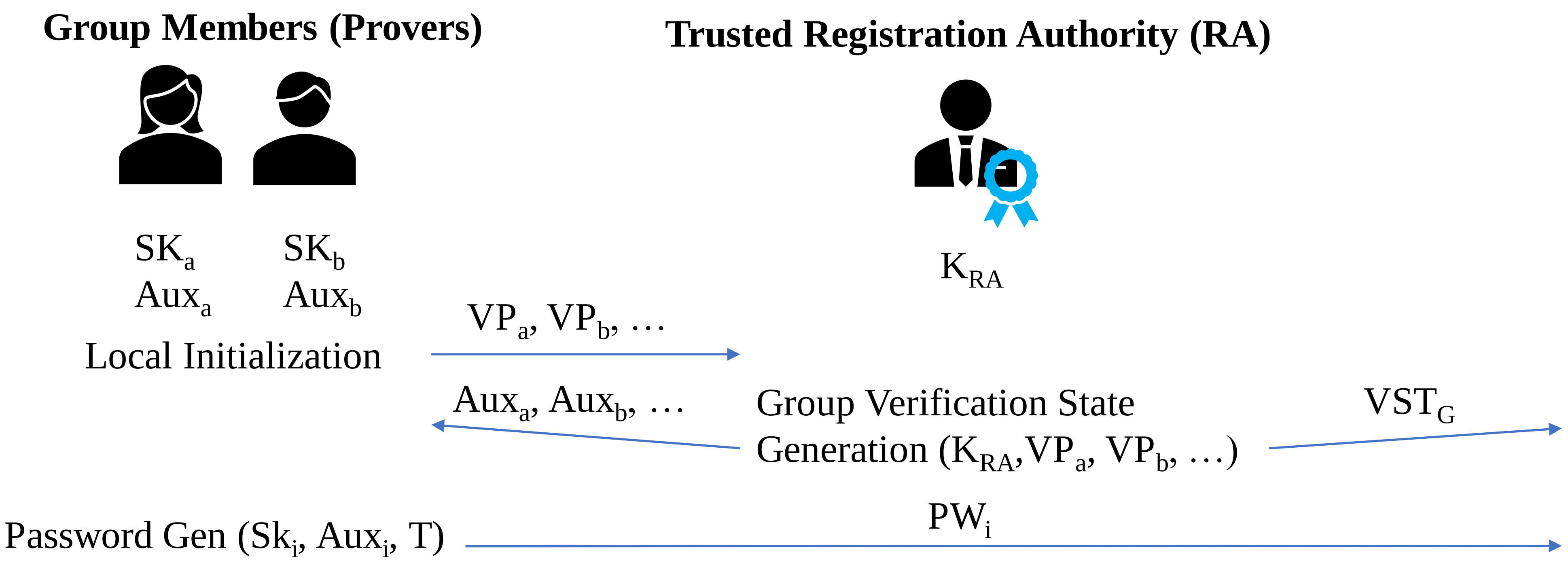

Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
SK_b
K_{RA}
VST_G
VP_a, VP_b, …
Local Initialization
Aux_a, Aux_b, …
Group Verification State Generation (K_{RA},VP_a, VP_b, …)
VST_G

Group TOTP

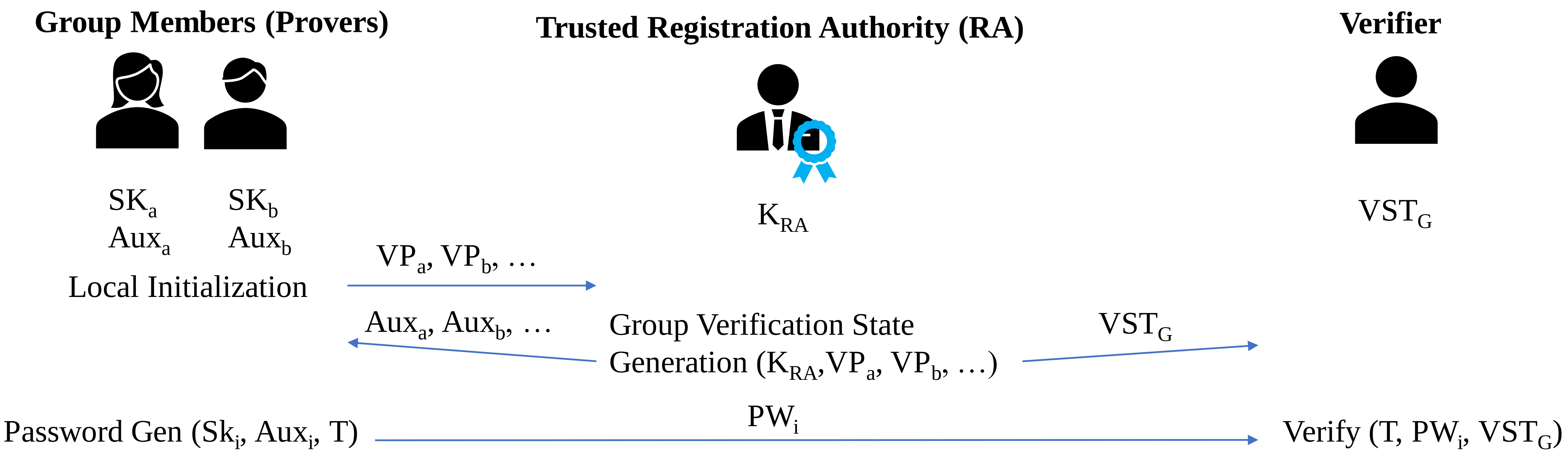


Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
Aux_a
SK_b
Aux_b
K_{RA}
VST_G
Local Initialization
VP_a, VP_b, …
Aux_a, Aux_b, …
Group Verification State
Generation (K_{RA},VP_a, VP_b, …)
VST_G

Group TOTP

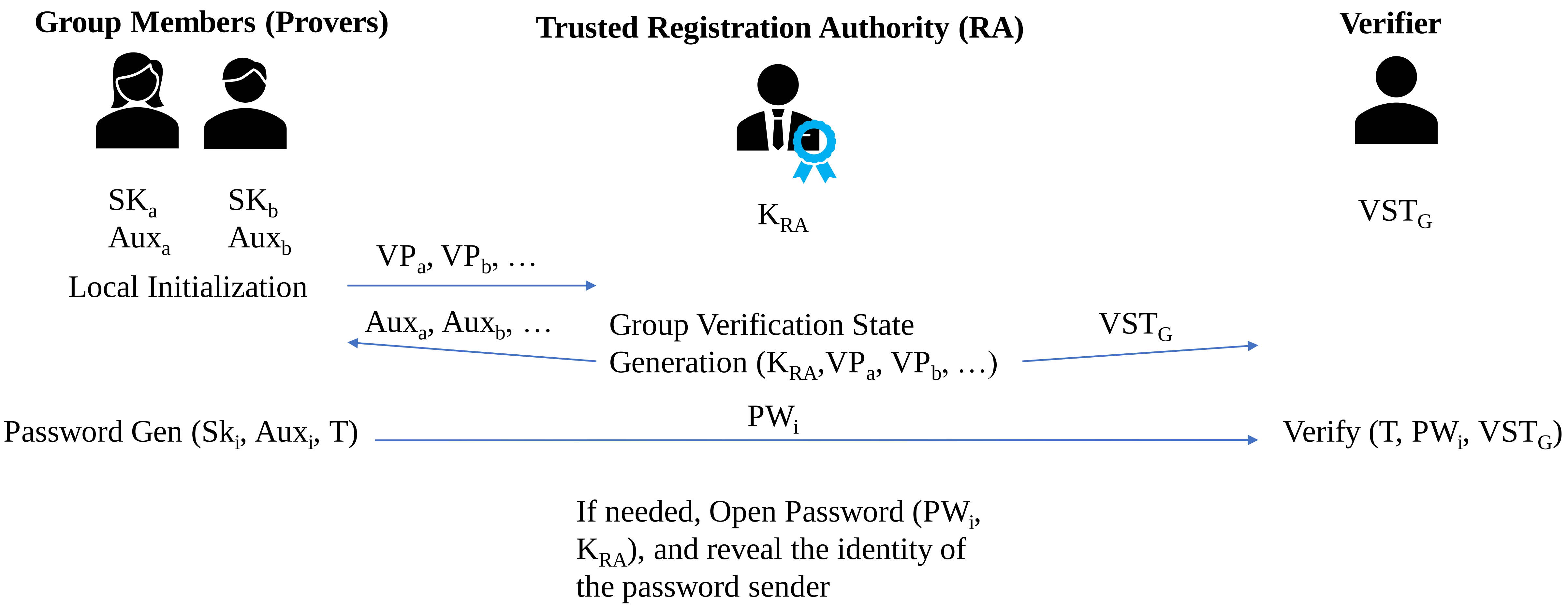

Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
Aux_a
SK_b
Aux_b
K_{RA}
VST_G
VP_a, VP_b, …
Local Initialization
Aux_a, Aux_b, …
Group Verification State Generation (K_{RA}, VP_a, VP_b, …)
VST_G
PW_i
Password Gen (Sk_i, Aux_i, T)

Group TOTP



Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK_a
Aux_a
SK_b
Aux_b
K_{RA}
VST_G
Local Initialization
VP_a, VP_b, …
Aux_a, Aux_b, …
Group Verification State Generation (K_{RA}, VP_a, VP_b, …)
VST_G
Password Gen (Sk_i, Aux_i, T)
PW_i
Verify (T, PW_i, VST_G)

Group TOTP


Group Members (Provers)
Trusted Registration Authority (RA)
Verifier
SK$_a$ Aux$_a$
SK$_b$ Aux$_b$
K$_{RA}$
VST$_G$
Local Initialization
VP$_a$, VP$_b$, …
Aux$_a$, Aux$_b$, …
Group Verification State Generation (K$_{RA}$,VP$_a$, VP$_b$, …)
VST$_G$
Password Gen (Sk$_i$, Aux$_i$, T)
PW$_i$
Verify (T, PW$_i$, VST$_G$)
If needed, Open Password (PW$_i$, K$_{RA}$), and reveal the identity of the password sender

Security Properties

Security Properties

- **Traceability**: adversary cannot create a password associated with an uncompromised secret seed of an uncorrupted member, such that the password is **valid** but **cannot be opened as associated with** the corresponding member

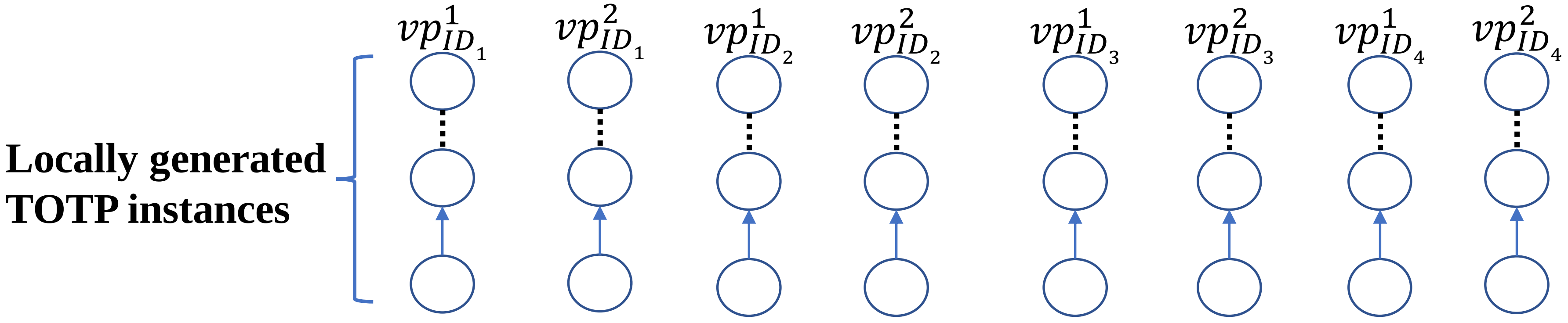
Security Properties

- **Traceability**: adversary cannot create a password associated with an uncompromised secret seed of an uncorrupted member, such that the password is **valid** but **cannot be opened as associated with** the corresponding member

- **Anonymity**: adversary cannot distinguish one group member's password from another's

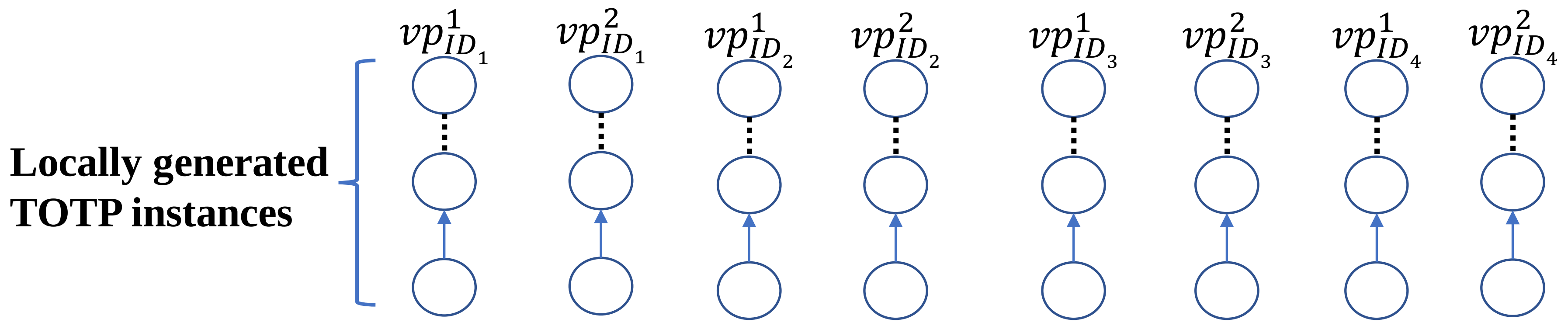
Detailed Construction of VST_G Generation

Detailed Construction of VST_G Generation

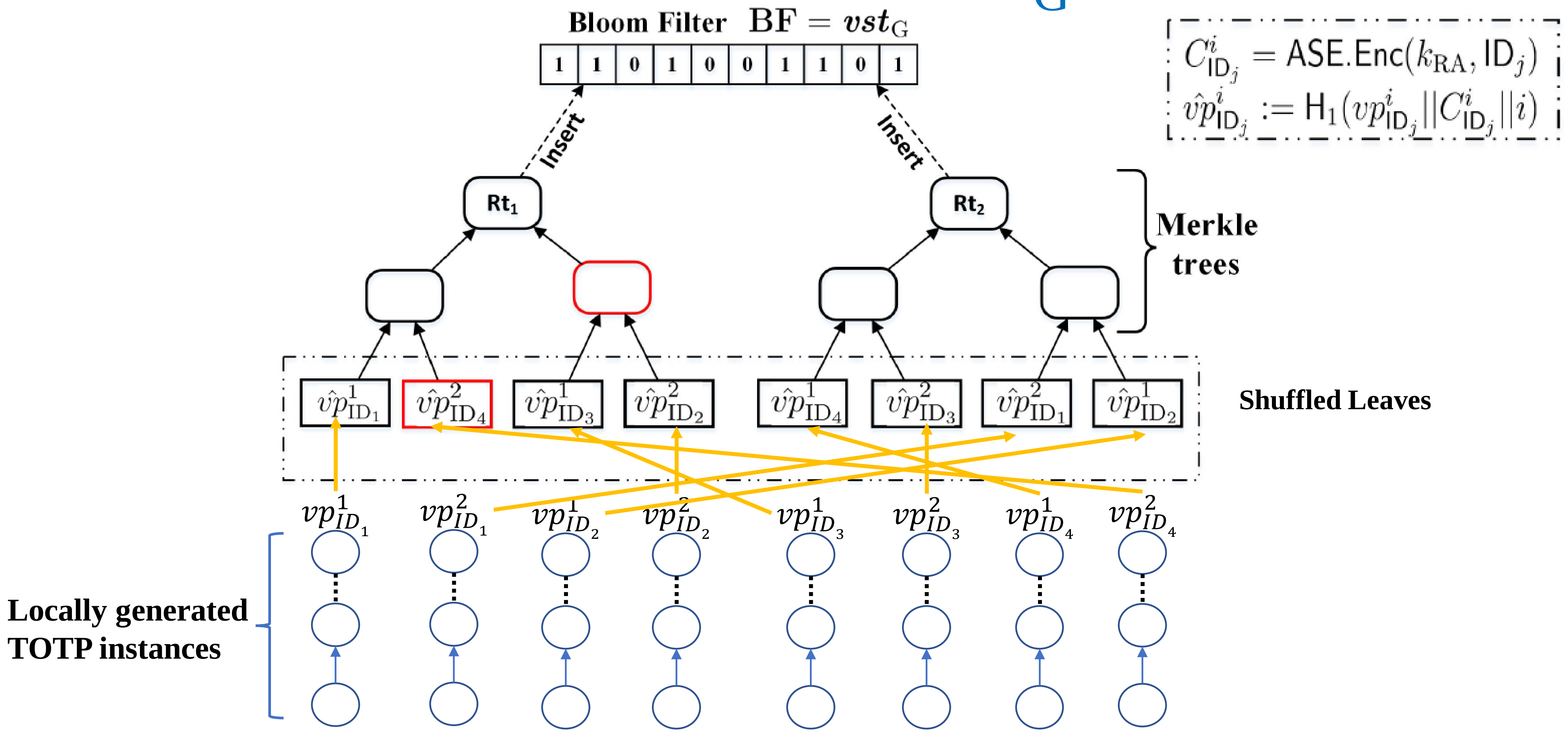


Detailed Construction of VST_G Generation

$$C^i_{\mathsf{ID}_j} = \mathsf{ASE.Enc}(k_{\mathrm{RA}}, \mathsf{ID}_j)$$
$$\hat{vp}^i_{\mathsf{ID}_j} := \mathsf{H}_1(vp^i_{\mathsf{ID}_j} || C^i_{\mathsf{ID}_j} || i)$$

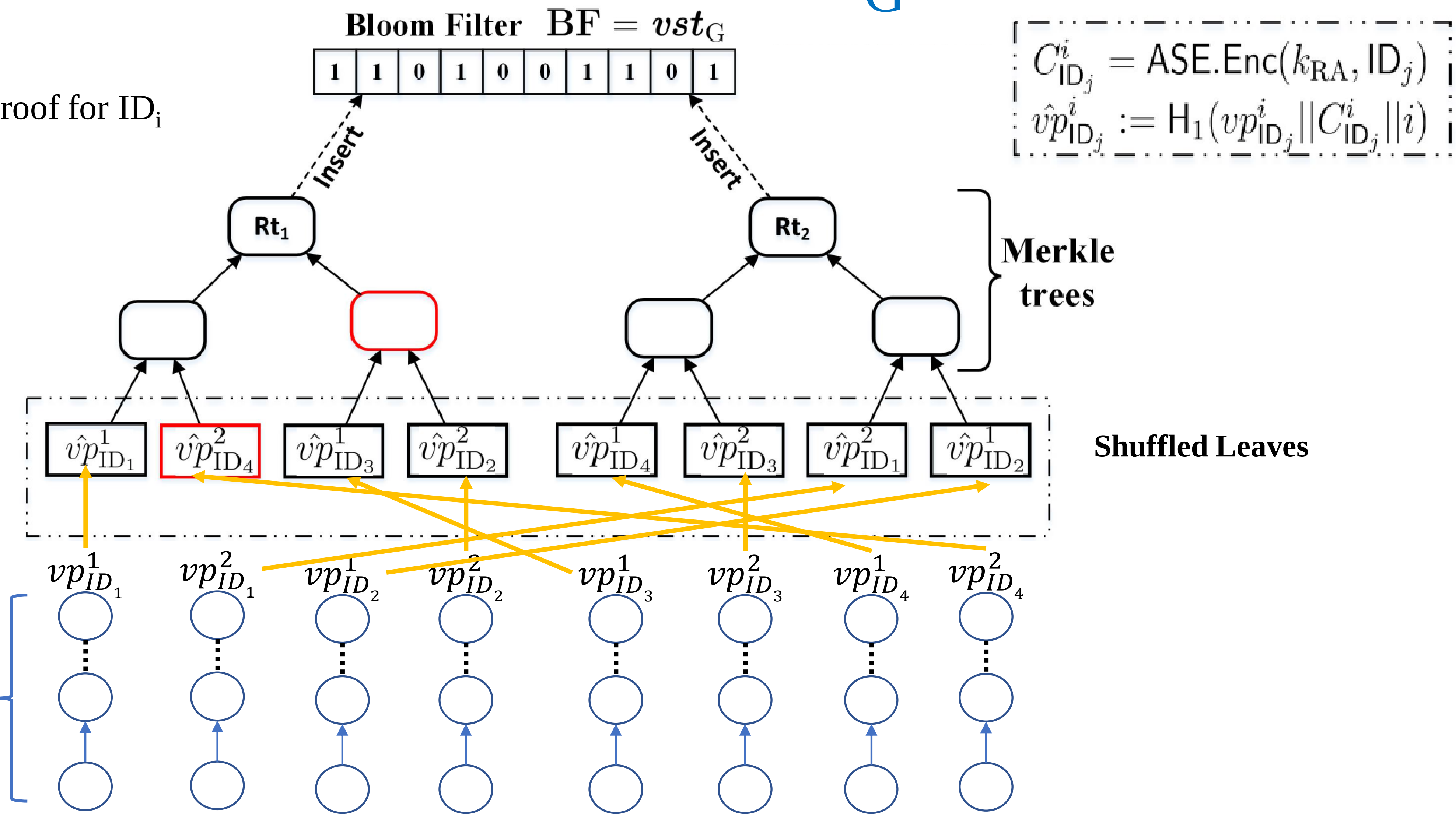


Detailed Construction of VST_G Generation



Detailed Construction of VST_G Generation

Aux_i = Merkle Proof for ID_i



Privacy-Preserving Proof of Location

Privacy-Preserving Proof of Location

- **Goal**: user proves where she/he was
  - allows users to record authenticated location data at times of their choice by presenting a fraud-proof location claim, without revealing the identities of protocol participants

Privacy-Preserving Proof of Location

- **Goal**: user proves where she/he was
  - allows users to record authenticated location data at times of their choice by presenting a fraud-proof location claim, without revealing the identities of protocol participants
- **Parties:**
  - **Registration Authority**: register for prover and witnesses
  - **Prover**: prove she/he was at a location at time T
  - **Witness**: testify the location of the prover based on its own location
  - **Verifier**: verify the location proofs
  - **Public Ledger**: record the location proofs and incentivize the witnesses

Privacy-Preserving Proof of Location

- **Goal**: user proves where she/he was
  - allows users to record authenticated location data at times of their choice by presenting a fraud-proof location claim, without revealing the identities of protocol participants
- **Parties:**
  - **Registration Authority**: register for prover and witnesses
  - **Prover**: prove she/he was at a location at time T
  - **Witness**: testify the location of the prover based on its own location
  - **Verifier**: verify the location proofs
  - **Public Ledger**: record the location proofs and incentivize the witnesses
- **Additional Building blocks:**
  - Commitment Scheme
  - Privacy-Preserving Location Proximity (PPLP) Scheme

Privacy-Preserving Proof of Location

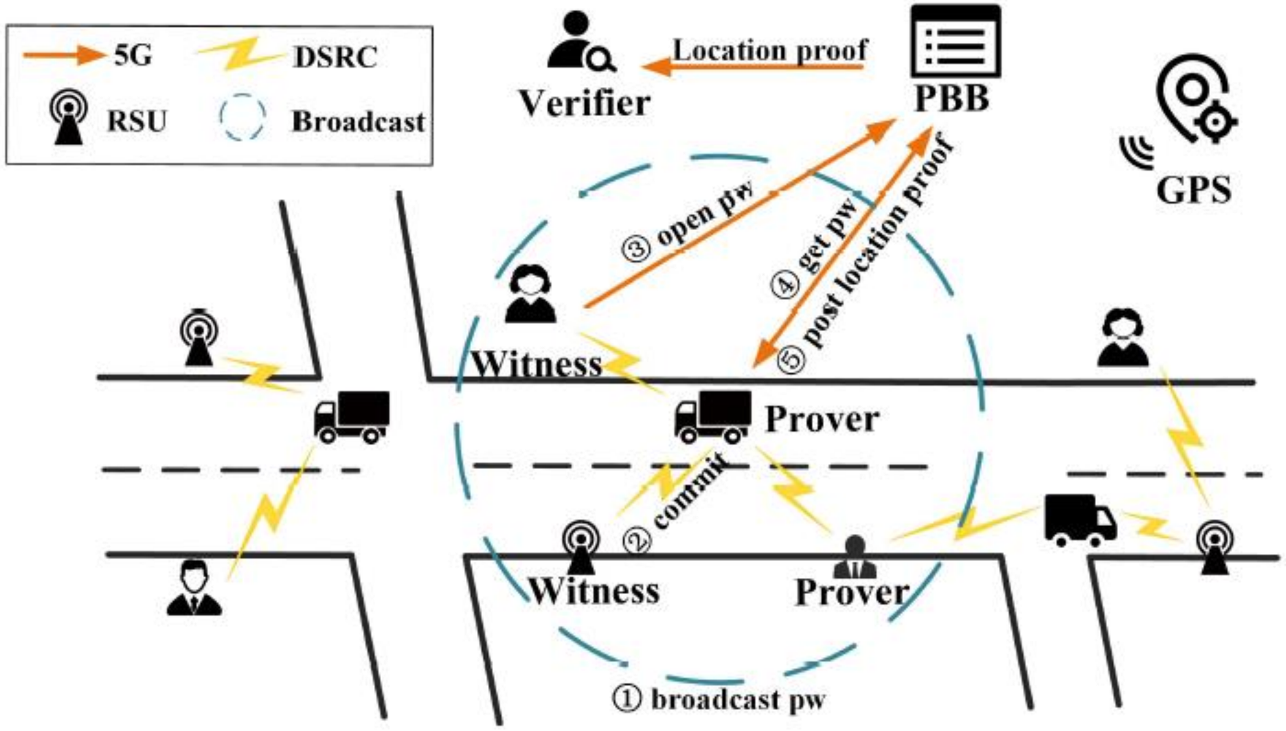
① A prover broadcasts its GTOTP password and privacy-preserving location proximity (PPLP) request to nearby witnesses via a short-range communication channel.

② Witnesses who can testify for the prover will respond with both message and location commitments regarding the PPLP responses.

③ Witnesses and prover exchange the password for verifying the message commitment.

④ The prover finally assembles the location proof based on the gathered proofs and publishes it to Public Ledger.

⑤ The verifier can obtain the location proof from either the Public Ledger or the prover.



Performance Evaluation

- Prover/witness: RPi3
- Verifier: PC with i7 CPU and 2GB RAM
- More detailed breakdown analysis in the paper

M	Computation time (s)				PfSize (KB)
	PfGen			Verify	
	Prover	Witness	Total	Verifier	
5	0.116/0.133	0.089/0.098	0.205/0.231	0.00065	1.16
10	0.237/0.276	0.089/0.098	0.326/0.347	0.0011	2.17
15	0.331/0.382	0. 0.089/0.098	0.42/0.48	0.0018	3.19

Performance Evaluation

- Prover/witness: RPi3
- Verifier: PC with i7 CPU and 2GB RAM
- More detailed breakdown analysis in the paper

M	Computation time (s)				PfSize (KB)
	PfGen			Verify	
	Prover	Witness	Total	Verifier	
5	0.116/0.133	0.089/0.098	0.205/0.231	0.00065	1.16
10	0.237/0.276	0.089/0.098	0.326/0.347	0.0011	2.17
15	0.331/0.382	0. 0.089/0.098	0.42/0.48	0.0018	3.19

Performance Evaluation

- Prover/witness: RPi3
- Verifier: PC with i7 CPU and 2GB RAM
- More detailed breakdown analysis in the paper

M	Computation time (s)				PfSize (KB)
	PfGen			Verify	
	Prover	Witness	Total	Verifier	
5	0.116/0.133	0.089/0.098	0.205/0.231	0.00065	1.16
10	0.237/0.276	0.089/0.098	0.326/0.347	0.0011	2.17
15	0.331/0.382	0. 0.089/0.098	0.42/0.48	0.0018	3.19

Summary

Summary

- Extend traditional TOTP to a group setting

Summary

- Extend traditional TOTP to a group setting
- Propose an efficient GTOTP construction

Summary

- Extend traditional TOTP to a group setting
- Propose an efficient GTOTP construction
- Demonstrate an application of GTOTP in privacy-preserving proof of location

Summary

- Extend traditional TOTP to a group setting
- Propose an efficient GTOTP construction
- Demonstrate an application of GTOTP in privacy-preserving proof of location

- Open question:
  - Dynamic group management