Distributed and Anonymous Authentication for Unstructured P2P Networks

Group 13

E/15/350: Tennakoon T.M.P.B.
E/15/180: Karunathilaka V.M.B.S.S.V

Introduction

Authentication

- To make one person trust another one
- Who is talking to whom must be as valid as whom he/she claims to be
  - Is he/she the valid person who is searching a public database?
  - Is he/she the valid person who provide you a movie without virus?

Anonymity and Privacy

- The right to be let alone.
- Who is talking to whom should be confidential or private in the Internet.
  - Who is searching a public database?
  - Which movie are you downloading?

Tradeoff: Anonymity vs Authentication

- Anonymity hides accountability.
    - No fear of being identified.
    - No responsibility for actions.
- Authentication provide accountability.
    - Responsibility for actions.

Challenge: Anonymous authentication in P2P

- Lack of Authentication.
- Misbehaving entities
- Uncontrolled anonymity.

Distributed & Anonymous Authentication for Unstructured P2P Networks

Related Works

Background

Publication	Similarities	Differences
Pseudo Trust	● Complete system ● Zero knowledge proofs to authenticate	● Pseudonyms to hide identity
CST	● Shamir's secret sharing to distribute a key.	● Collaboration signatures to hide identity. ● Use of reputation management systems
Fair Blind Signature Based Authentication	● Shamir's secret sharing to distribute a key.	● FBST to hide identity ● Use of reputation management systems
Authentication with controlled anonymity in P2P systems	● Zero knowledge proofs to authenticate	● Merkle's puzzles to share a secret
PPAA	● Zero knowledge proofs to authenticate	● Tags to hide identity

Cryptographic Primitives

Zero Knowledge Proof

- Prove the possession of some secret **without revealing any information related to the secret**.

- We utilize a Schnorr's non interactive zero knowledge proof.

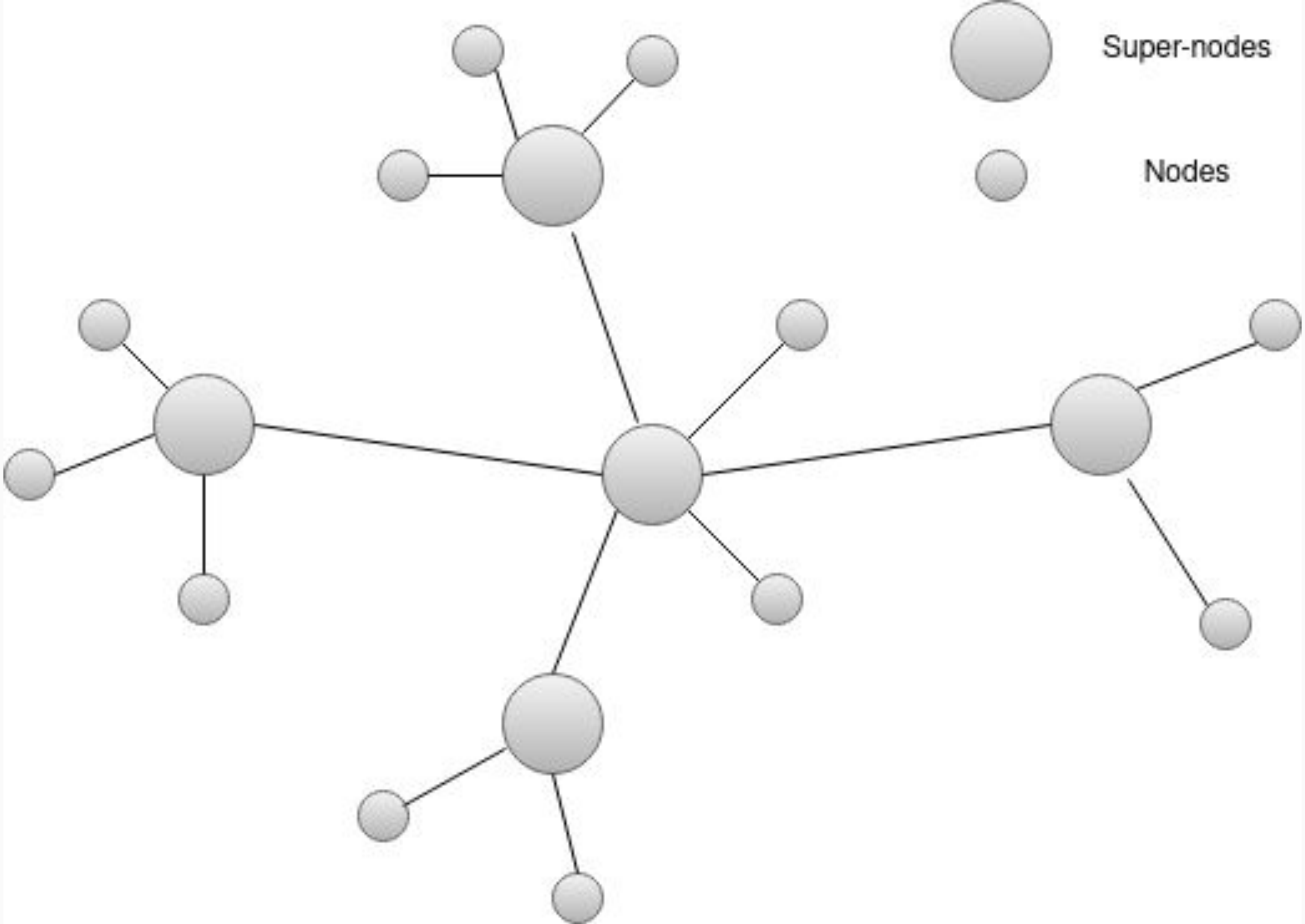
Ring Signatures

- Sign a message behalf of a group.
- Prove signer is a member of a group.
- **Infeasible to find exactly which member.**

Shamir's Secret Sharing

- Divide a secret into parts.
- **Reconstruct the original secret with a subset of the parts.**

Network Design


Super-nodes
Nodes

Distributed Certificate Management

Challenges in Certificate Management in P2P

- Absence of a central storage location.
- Super peers can leave any time.

Solution: Shamir’s Secret Sharing

1. Break the certificate into n parts.

2. Distributed the parts across the network.

3. Request the parts when needed

4. Reconstruct the certificate using r parts ($1 < r < n$)

Advantages

- Size of each part does not is exceed the certificate.
- Only require r parts to reconstruct the certificate.
- Flexible

Demo

Anonymous Authentication Protocols

1. Ring Signature based approach
2. Key Sharing based approach
3. Zero Knowledge Proof based approach

Ring Signature based approach

Prover		Verifier
Randomly select n-1 certificates and generate $C = \{Cert_1, Cert_2, ..., Cert_p, .., Cert_n\}$		
Generate $P = \{P_1, P_2, ..., P_p, .., P_n\}$		
Generate $M1 = Enc_{pv}(C)$		
	M1 $\rightarrow$	
		$C = Dec_{Sv}(M1)$
		Generate $P = \{P_1, P_2, ..., P_p, .., P_n\}$
		Generate $H = Hash(P)$
		Generate random number N
	$\leftarrow$ H, N	
Generate $H' = Hash(P)$		
Check if $H = H'$, if not terminate, otherwise continue		
Use secret key S_p, P, P_s and N generate ring signature σ		
Generate $M2 = Enc_{pv}(\sigma)$		
	M2 $\rightarrow$	
		Generate $\sigma = Dec_{sv}(M2)$
		Check if σ verifies to N using P, if not send Fail message, otherwise authentication is successful

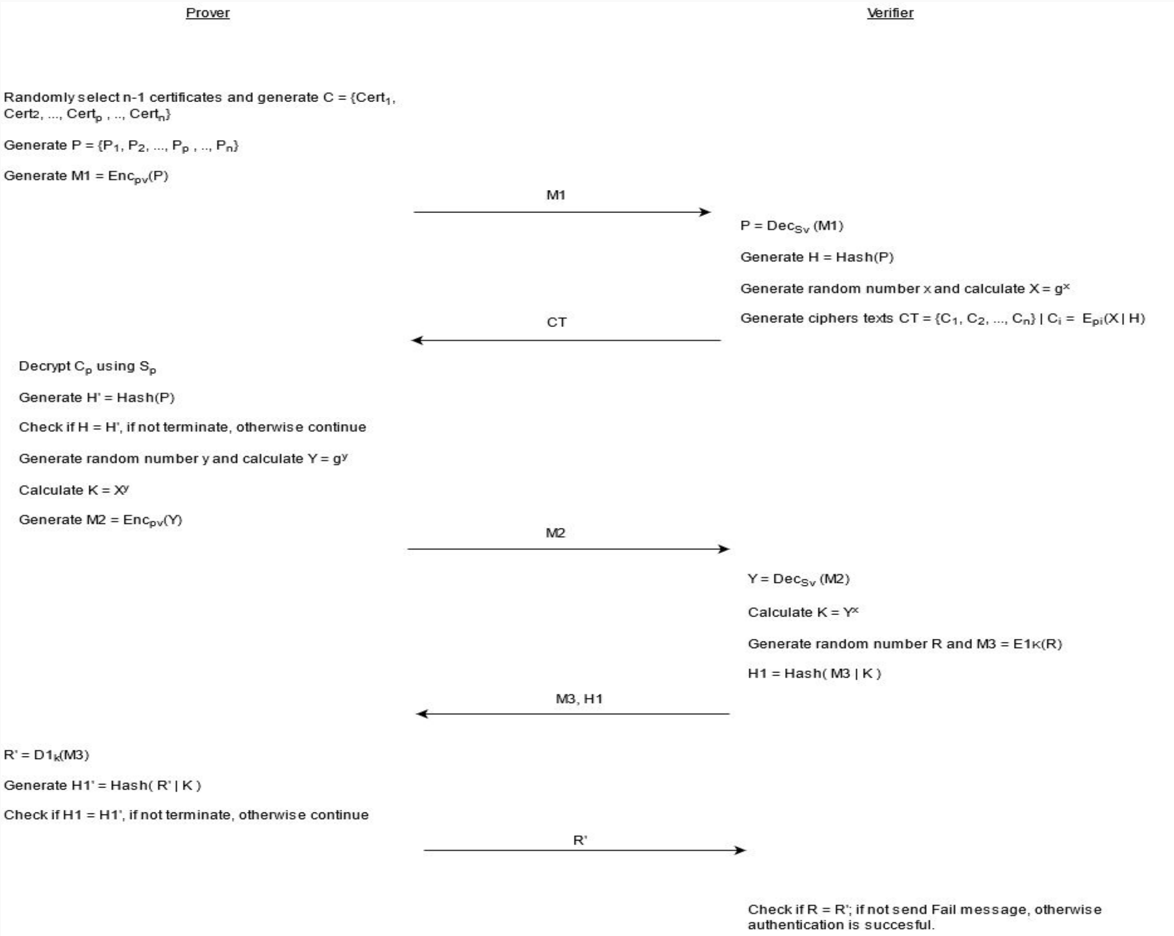
Summary

- Prover collects a set of certificates and shuffle his certificate into the set.

- **Verifier generate a random nonce and challenge prover to generate a ring signature using the set of certificates.**

- The knowledge of atleast one private key is required to generate the ring signature.

Key Sharing based approach



Summary

- Prover collects a set of certificates and shuffle his certificate into the set.
- Verifier generate a random number and encrypt with each public key.
- Challenge prover to generate a shared key using this random number.

Summary

- Prove need to decrypt any of the cipher texts to generate the shared key.

- The knowledge of atleast one private key is required to decrypt any of the cipher text.

Zero Knowledge Proof base approach

Prover | **Verifier**

Randomly select n-1 certificates and generate the corresponding public key set $P = \{A_1, A_2, \ldots, A_{n-1}\}$

Select a random number $s \mid s \in [0, Q\text{-}1]$

Select n-1 random numbers to generate $V = \{v_1, v_2, \ldots, v_{n-1}\} \mid v_i$ is a random number and $v_i \in [0, Q\text{-}1]$

Calculate $U = g^s A^{v1} A^{v2} \ldots A^{vn-1}$

$$\xrightarrow{\quad U \quad}$$

Select a random number $c \mid c \in [0, Q\text{-}1]$

$$\xleftarrow{\quad c \quad}$$

Calculate $v_p = v_1 \oplus v_2 \oplus \ldots v_{n-1} \oplus c$

Update $V = \{v_1, v_2, \ldots v_p, \ldots, v_{n-1}\}$

Update $C = \{Cert_1, \ldots, Cert_p, \ldots, Cert_{n-1}\}$

Calculate $r = s - a_p v_p \bmod p$

$$\xrightarrow{\quad r, V, C \quad}$$

Calculate $c' = v_1 \oplus v_2 \oplus \ldots v_n$

Check if $c = c'$; if not send Fail message, otherwise continue

Calculate $U' = g^r A^{v1} A^{v2} \ldots A^{n-1}$

Check if $U = U'$; if not send Fail message, otherwise authentication is succesful.

Group Parameters

- P, Q : Large Prime Numbers (Q | P - 1)
- a_u : Private Key ($a_u \in [1, Q - 1]$)
- $A_u = g^{au} \bmod P$: Public Key

Prover | Verifier

Randomly select n-1 certificates and generate the corresponding public key set $P = \{A_1, A_2, \ldots, A_{n-1}\}$

Select a random number s | $s \in [0, Q-1]$

Select n-1 random numbers to generate $V = \{v_1, v_2, \ldots, v_{n-1}\}$ | v_i is a random number and $v_i \in [0, Q-1]$

Calculate $U = g^s A^{v1} A^{v2} \ldots A^{vn-1}$

$\xrightarrow{\quad U \quad}$

Zero Knowledge Proof based approach

Zero Knowledge Proof based approach

Prover		Verifier
Randomly select n-1 certificates and generate the corresponding public key set $P = \{A_1, A_2, \ldots, A_{n-1}\}$		
Select a random number $s \mid s \in [0, Q-1]$		
Select n-1 random numbers to generate $V = \{v_1, v_2, \ldots, v_{n-1}\} \mid v_i$ is a random number and $v_i \in [0, Q-1]$		
Calculate $U = g^s A^{v1} A^{v2} \ldots A^{vn-1}$	$\xrightarrow{U}$	
		Select a random number $c \mid c \in [0, Q-1]$
	$\xleftarrow{c}$	

Prover		Verifier
Randomly select n-1 certificates and generate the corresponding public key set $P = \{A_1, A_2, \ldots, A_{n-1}\}$		
Select a random number $s \mid s \in [0, Q-1]$		
Select n-1 random numbers to generate $V = \{v_1, v_2, \ldots, v_{n-1}\} \mid v_i$ is a random number and $v_i \in [0, Q-1]$		
Calculate $U = g^s A^{v1} A^{v2} \ldots A^{vn-1}$		
	$\xrightarrow{U}$	
		Select a random number $c \mid c \in [0, Q-1]$
	$\xleftarrow{c}$	
Calculate $v_p = v_1 \oplus v_2 \oplus \ldots v_{n-1} \oplus c$		
Update $V = \{v_1, v_2, \ldots v_p, \ldots, v_{n-1}\}$		
Update $C = \{Cert_1, \ldots, Cert_p, \ldots, Cert_{n-1}\}$		
Calculate $r = s - a_p v_p \bmod p$		
	$\xrightarrow{r,\, V,\, C}$	

Zero Knowledge Proof based approach

Prover | Verifier

Prover

Randomly select n-1 certificates and generate the corresponding public key set $P = \{A_1, A_2, \ldots, A_{n-1}\}$

Select a random number s | $s \in [0, Q-1]$

Select n-1 random numbers to generate $V = \{v_1, v_2, \ldots, v_{n-1}\}$ | v_i is a random number and $v_i \in [0, Q-1]$

Calculate $U = g^s A^{v1} A^{v2} \ldots A^{vn-1}$

$\xrightarrow{U}$

Verifier

Select a random number c | $c \in [0, Q-1]$

$\xleftarrow{c}$

Prover

Calculate $v_p = v_1 \oplus v_2 \oplus \ldots v_{n-1} \oplus c$

Update $V = \{v_1, v_2, \ldots v_p, \ldots, v_{n-1}\}$

Update $C = \{Cert_1, \ldots, Cert_p, \ldots, Cert_{n-1}\}$

Calculate $r = s - a_p v_p \bmod p$

$\xrightarrow{r,\ V,\ C}$

Verifier

Calculate $c' = v_1 \oplus v_2 \oplus \ldots v_n$

Check if c = c'; if not send Fail message, otherwise continue

Calculate $U' = g^r A^{v1} A^{v2} \ldots A^{n-1}$

Check if U = U'; if not send Fail message, otherwise authentication is succesful.

Zero Knowledge Proof based approach

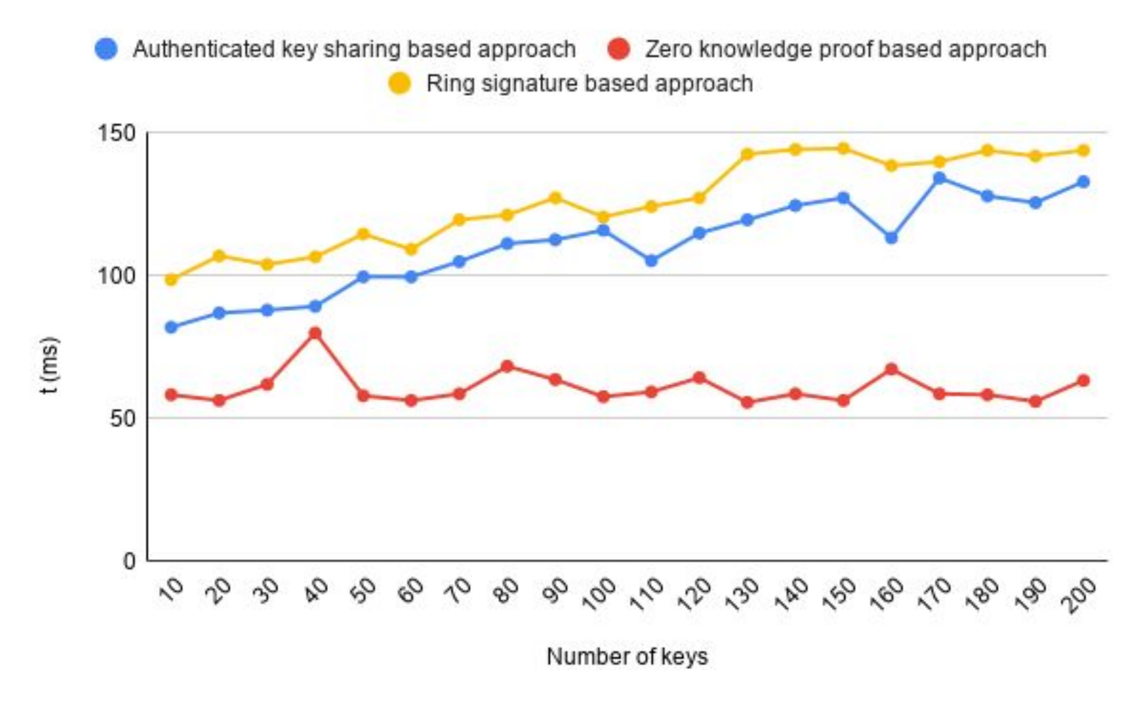
Security Analysis

Security Analysis

	Ring Signature base approach	Key Sharing based approach	Zero Knowledge proof based approach
Completeness	✓	✓	✓
Soundness	✓	✓	✓
impersonation	✓	✓	✓
Replay attacks	✓	✓	✓
K - anonymity	✓	X	✓

Performance Analysis

Performance of authentication protocols



Drawbacks

Drawbacks

- Reputation management system is incompatible with authentication protocol.

Conclusion and Future Works

- Modify the zero-knowledge proof-based approach for anonymity revocation
- Integrate the proposed authentication protocols in real-world peer to peer transactions

Background