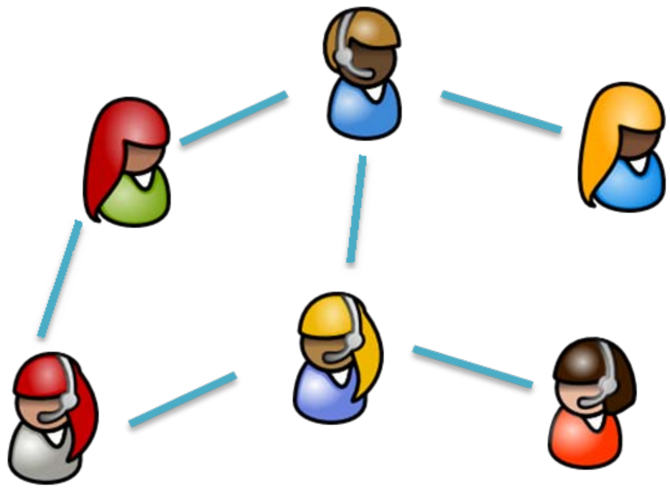
Privacy-Preserving Admission to Mobile P2P Groups

Mark Manulis
Cryptographic Protocols Group
TU Darmstadt & CASED

CRYPTOGRAPHIC PROTOCOLS

Scenario: Mobile P2P Groups

Goal Establishment of a (closed) p2p group by mobile users



Research questions

- How to build a group?
- How to admit new members?
- How to prove membership?
- How to communicate securely?

Technical constraints

- Decentralized infrastructure
- Mobility

Group Management Framework

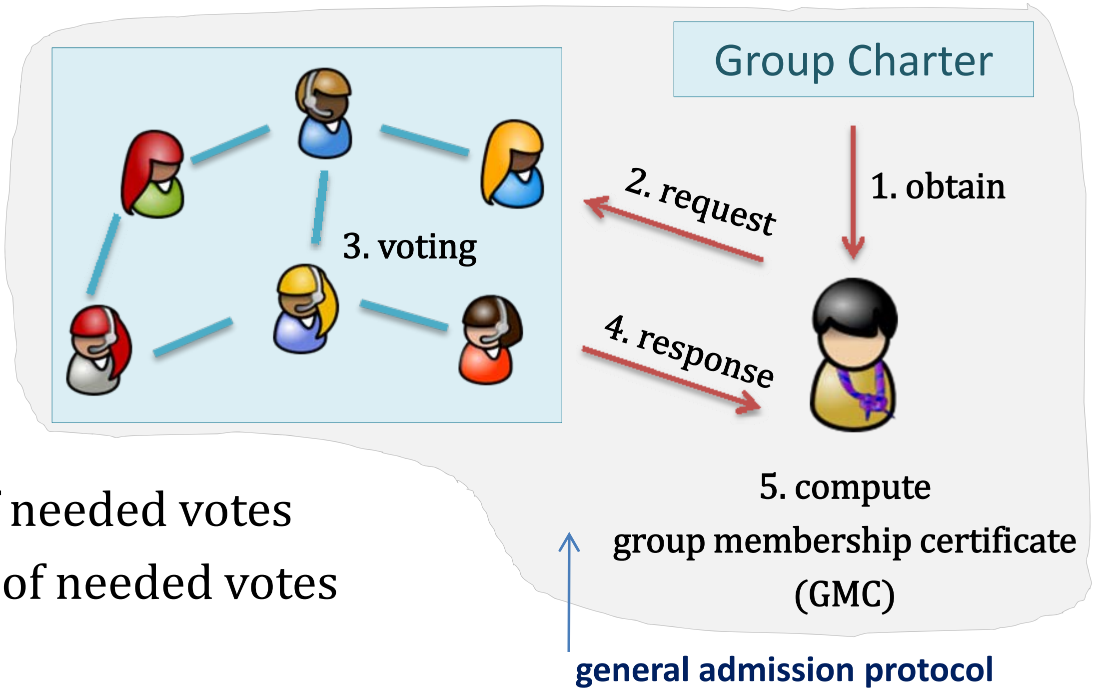
Kim-Mazzocchi-Tsudik Group Management Framework[KMT'03]

Group Charter contains public information about the group

Group Authority manages group admission, is either centralized or *distributed*

Admission Policy Types

- Access Control Lists
  not applicable to p2p groups
- Centralized decision
  not applicable to p2p groups
- **Collective decision (voting)**
  - static with fixed threshold of needed votes
  - *dynamic* with some fraction of needed votes



Prior Work uses Threshold Signatures

Digital Signatures

Key generation algorithm returns secret key sk and public key pk.

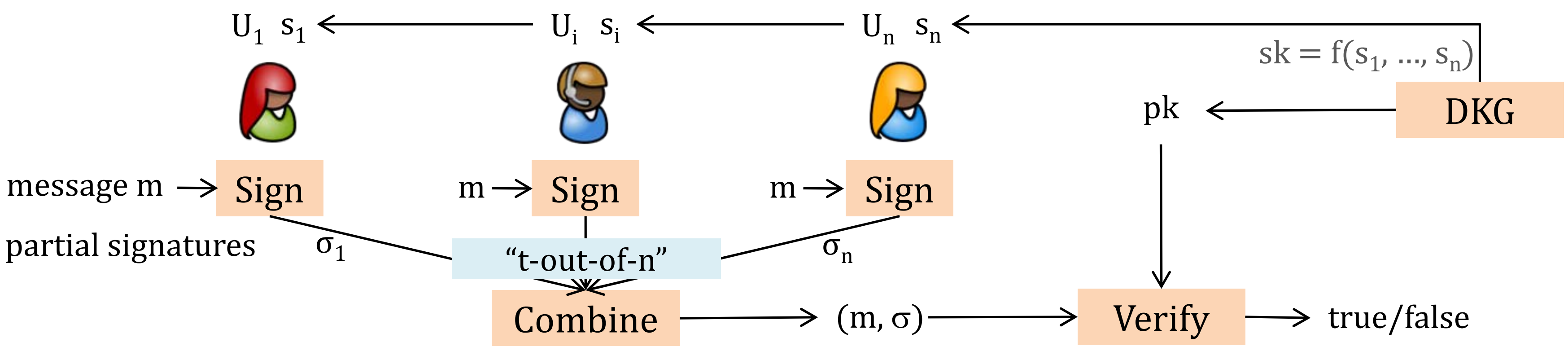
Signature σ on a message m can be computed using sk and verified using pk.

Threshold Signatures

Users run *distributed key generation (DKG)* and compute public key pk.

Each user U_i holds a *share* s_i of the secret key sk = $f(s_1, \dots, s_n)$. <u>sk remains unknown.</u>

Signature σ on a message m can be computed by at least „t-out-of-n“ users.



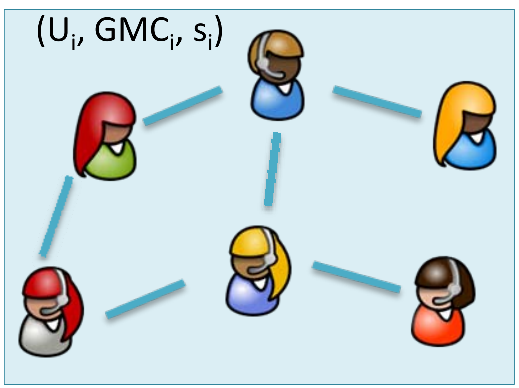
Threshold-Sig-based Admission Control

Admission Process (general for schemes in [NTY‘03, STY04, STY05])

- New member U* obtains pk and sends out own membership request.
- U* requires at least t votes to compute own membership certificate GMC*.
- Each vote gives a *partial signature* σ_i on the infos from membership request.
- Each vote gives *a partial share* $ps_i(U^*)$ allowing U* to compute own share s*.

initialisation is performed by founding group members

DKG → Group Charter: pk, t, U_1, …, U_n



1. obtain

2. request (U*)

3+4. at least t responses (U_i, GMC_i, σ_i, $ps_i(U^*)$)

5. compute

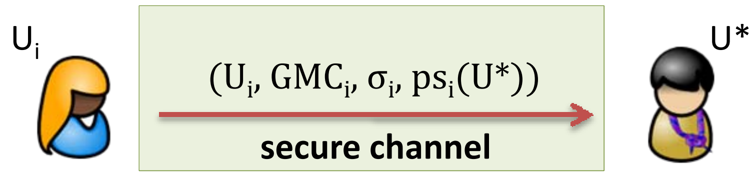
certificate $GMC^* = \sigma = f(\sigma_1, \ldots, \sigma_t)$

secret share $s^* = f(ps_1(U^*), \ldots, ps_t(U^*))$

Some Drawbacks

Need for Secure Channels

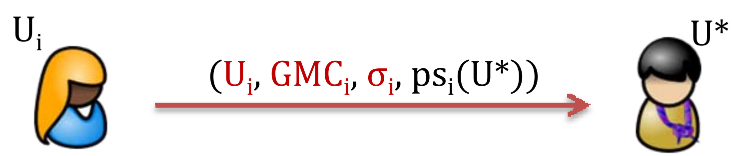
- Distribution of partial shares $ps_i(U^*)$ requires secure channels.
- Otherwise any eavesdropper would be able to compute the share s^*.



Need for Randomization of Shares

- Given $ps_i(U^*)$ it is possible for U^* to compute the secret share s_i of U_i.
- Avoiding this requires expensive *random shuffling*[HJKY'95].

Lack of Vote Privacy

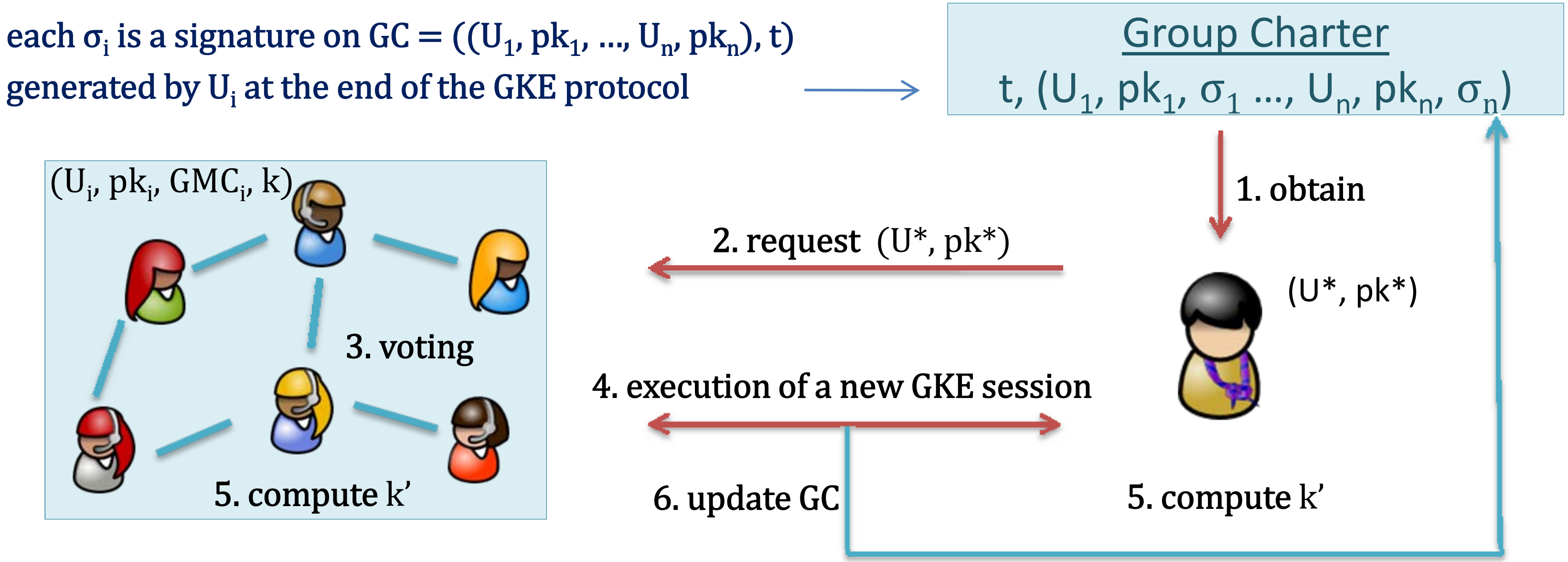


- Votes reveal identities of members.
- U^* learns who voted in favor of admission (or against it).

Overview of Our Approach

Admission Control based on Group Key Exchange (GKE)

- Founding users run Group Key Exchange and compute *shared key* k.
- U* sends own membership request to the group.
- All U_i vote *securely* within the group, i.e. encrypting their votes with k.
- If #(positive votes) > t then all U_i and U* execute new GKE and compute k‘.

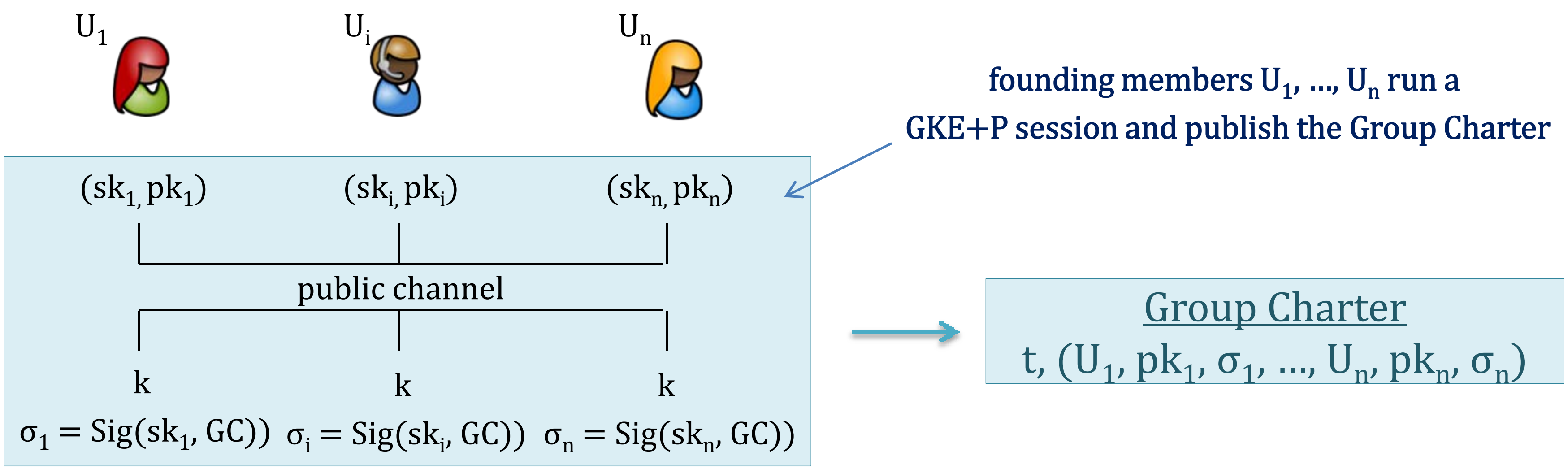


Initialization using a GKE+P Protocol

Group Key Exchange with On-Demand Derivation of P2P Keys (GKE+P) [M09, ACMP10]

- Computes the *group key* k and *p2p keys* $k_{i,j}$ shared between U_i and U_j only.
- Each U_i generates *ephemeral key pair* (sk_i, pk_i) during the protocol execution.
- Each generated ephemeral public key pk_i is bound to the GKE execution.

Initialization by Founding Group Members



where $GC = (t, (U_1, pk_1, ..., U_n, pk_n))$ and t is the *dynamic* fraction of votes

Voting Process

Voting Process by Current Group Members

- Each U_i holds group key k and own (sk_i, pk_i).
- (sk_i, pk_i) can be used to sign messages.

Group Charter

$t, (U_1, pk_1, \sigma_1, \ldots, U_n, pk_n, \sigma_n)$

1. obtain

$U_1, k, (sk_1, pk_1)$ $U_i, k, (sk_i, pk_i)$ $U_n, k, (sk_n, pk_n)$



2. request (U^*, pk^*)

(U^*, pk^*)

$vote_i = acc/dec$

$\sigma_i = Sig(sk_i, (vote_i, U^*, pk^*, GC))$

$c_i = Enc(k, (vote_i, U^*, pk^*, GC, \sigma_i))$

c_1 c_i c_n

public channel

decrypt all c_j

verify all σ_j using pk_j

eliminate *incorrect* votes

compute fraction t

encryption with k prevents U* from learning votes

– pk_j are taken from GC, thus indicating valid members

– using signatures we further prevent double voting

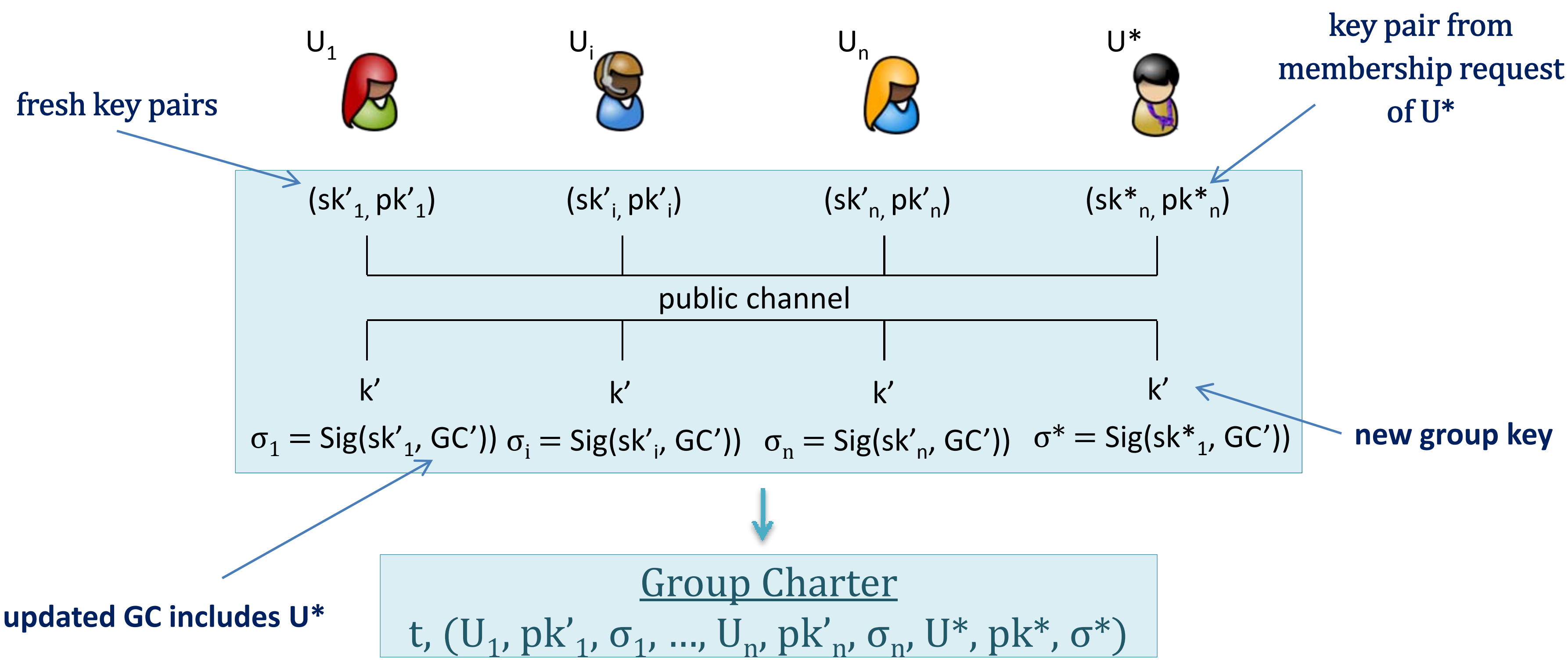
if t is sufficient

then execute GKE+P

Admission to the Group

Admission of U* to the Group

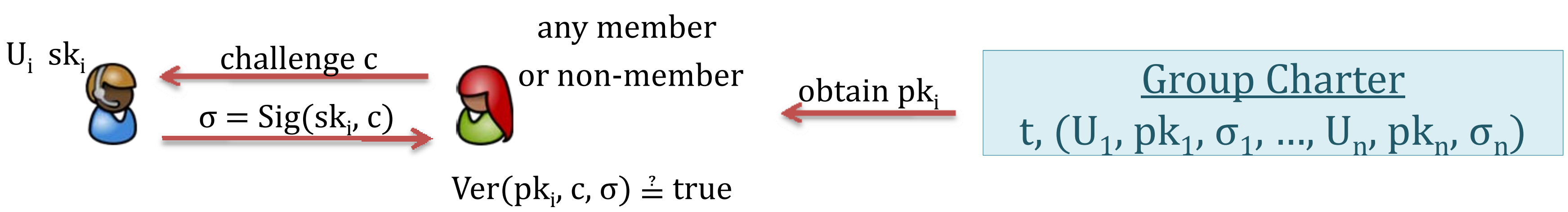
- the protocol proceeds similar to the initialization step
- all users including U* participate in the GKE+P session



Proving Own Group Membership

Proving Group Membership to Insiders and Outsiders

- U_i's public key pk_i is included in GC and signed by all other members
- U_i can prove own membership in a simple signature-based challenge-response



Proving Group Membership without Disclosing own Identity

- U_i can run a *zero-knowledge proof of knowledge*
- U_i proves knowledge of 1-out-of-n private keys sk_i w/o disclosing the exact pk_i
- e.g. using dlog based $(sk_i, pk_i) = (x, g^x)$ one can use the proof from [CM98]

Various Forms of Secure Communication

Secure Group Communication

- members can communicate securely within the group using the *group key* k

Secure P2P Communication

- GKE+P allows any pair of users U_i and U_j to derive a *p2p key* $k_{i,j}$
- this derivation does not require any additional communication
- U_i and U_j can use $k_{i,j}$ to exchange secure messages
- $k_{i,j}$ remains secret from other parties (including other members)

Secure Communication with Outsiders

- any non-member can encrypt messages for any U_i using pk_i from GC
- group key k can be used to derive a group key pair (sk_G, pk_G) such that any outsider can send encrypted message to the whole group using pk_G

Security Issues

Unforgeability

- the goal is to prevent adversary $\mathcal{A}$ from claiming group membership
- in our solution membership can be claimed via an execution of the challenge-response protocol using (Sig, Ver) and public key pk_i from GC
- note that each member's public key pk_i is signed by *all* other members
- $\mathcal{A}$ cannot claim group membership since the signature is unforgeable

Anonymity (as a new goal)

- applies only to admissions based on collective decisions
- the goal is to prevent adversary $\mathcal{A}$ from learning (U_i, $vote_i$)
- in our solution votes are exchanged encrypted with the group key k
- the group key k remains secret from $\mathcal{A}$ due to security of GKE+P
- all (U_i, $vote_i$) remain secret from $\mathcal{A}$ due to the security of (Enc, Dec)

Conclusion

Group Admission Protocols

- anonymity as a new privacy threat in admission control protocols for p2p groups
- current solutions based on threshold signatures do not support vote privacy

Solution based on GKE+P protocols

- users jointly initialize the group through the run of the GKE+P protocol
- dynamic admission policy is achieved via voting
- voting process preserves privacy of votes
- group membership can be easily proven with challenge-response techniques (possibly without disclosing the identity of the member)

Secure (Intra- and Intergroup) Communications

- secure group communication inside the group and with the outsiders
- secure p2p communication between group members and with outsiders
- flexible GKE protocol from [ACMP10] allows communication within subgroups