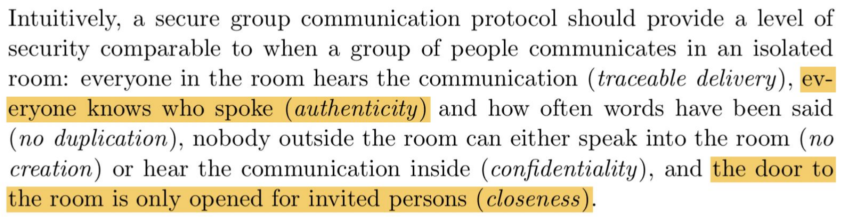
Authentication


Intuitively, a secure group communication protocol should provide a level of security comparable to when a group of people communicates in an isolated room: everyone in the room hears the communication (*traceable delivery*), everyone knows who spoke (*authenticity*) and how often words have been said (*no duplication*), nobody outside the room can either speak into the room (*no creation*) or hear the communication inside (*confidentiality*), and the door to the room is only opened for invited persons (*closeness*).

Objectives

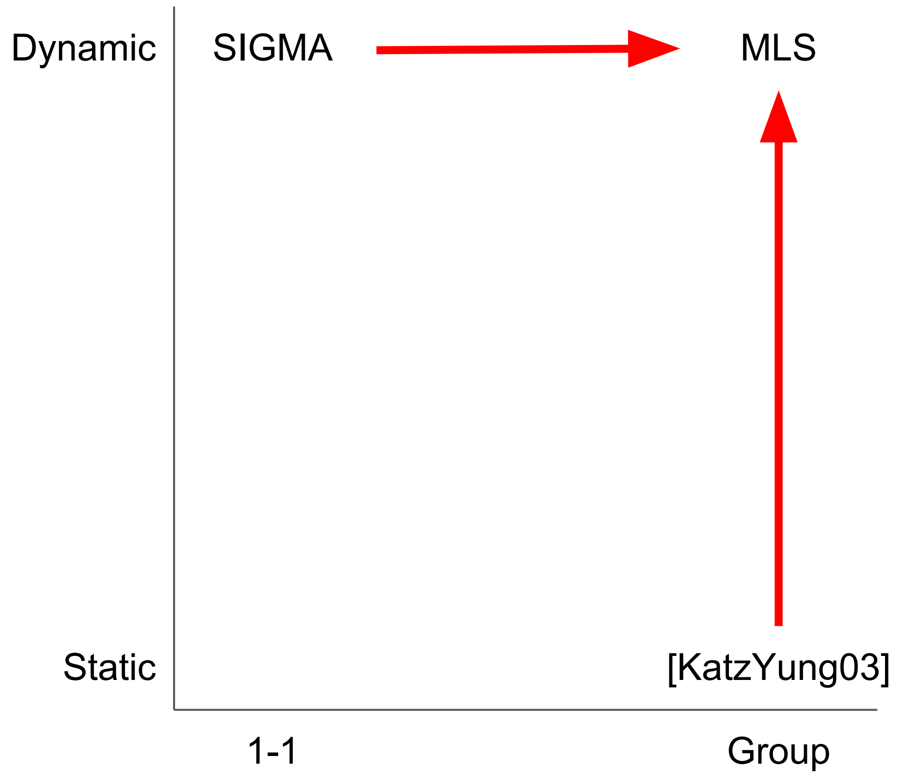
Add/Remove: Verify that sender is authorized to perform Add/Remove

Update: Verify that sender is legitimate holder of the slot being updated

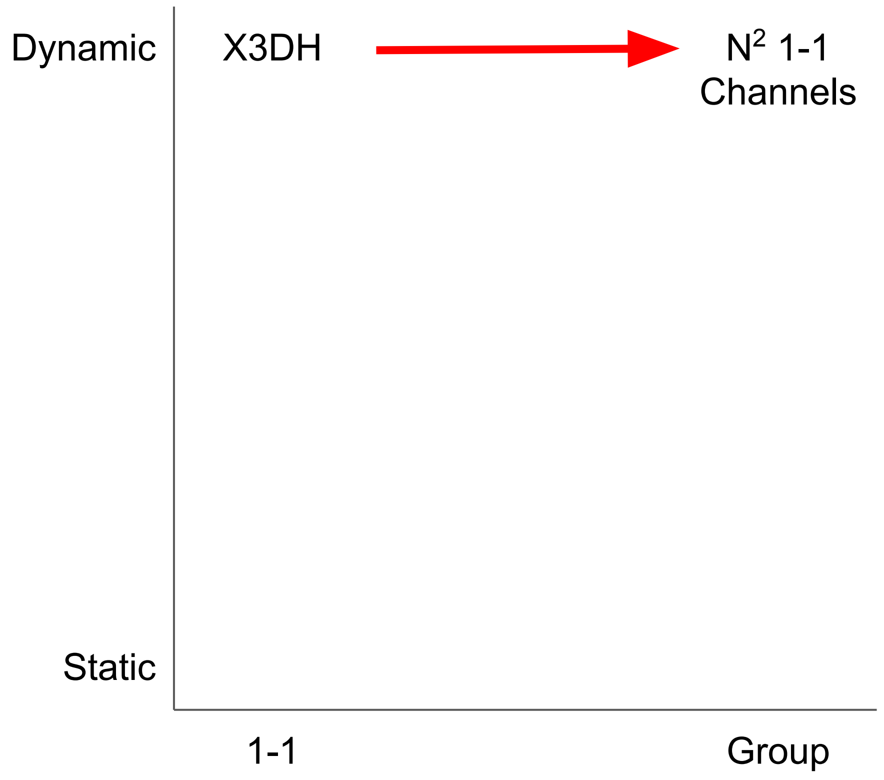
Application/Send: Verify the identities of the parties who can decrypt

Application/Recv: Verify the identity of the sender

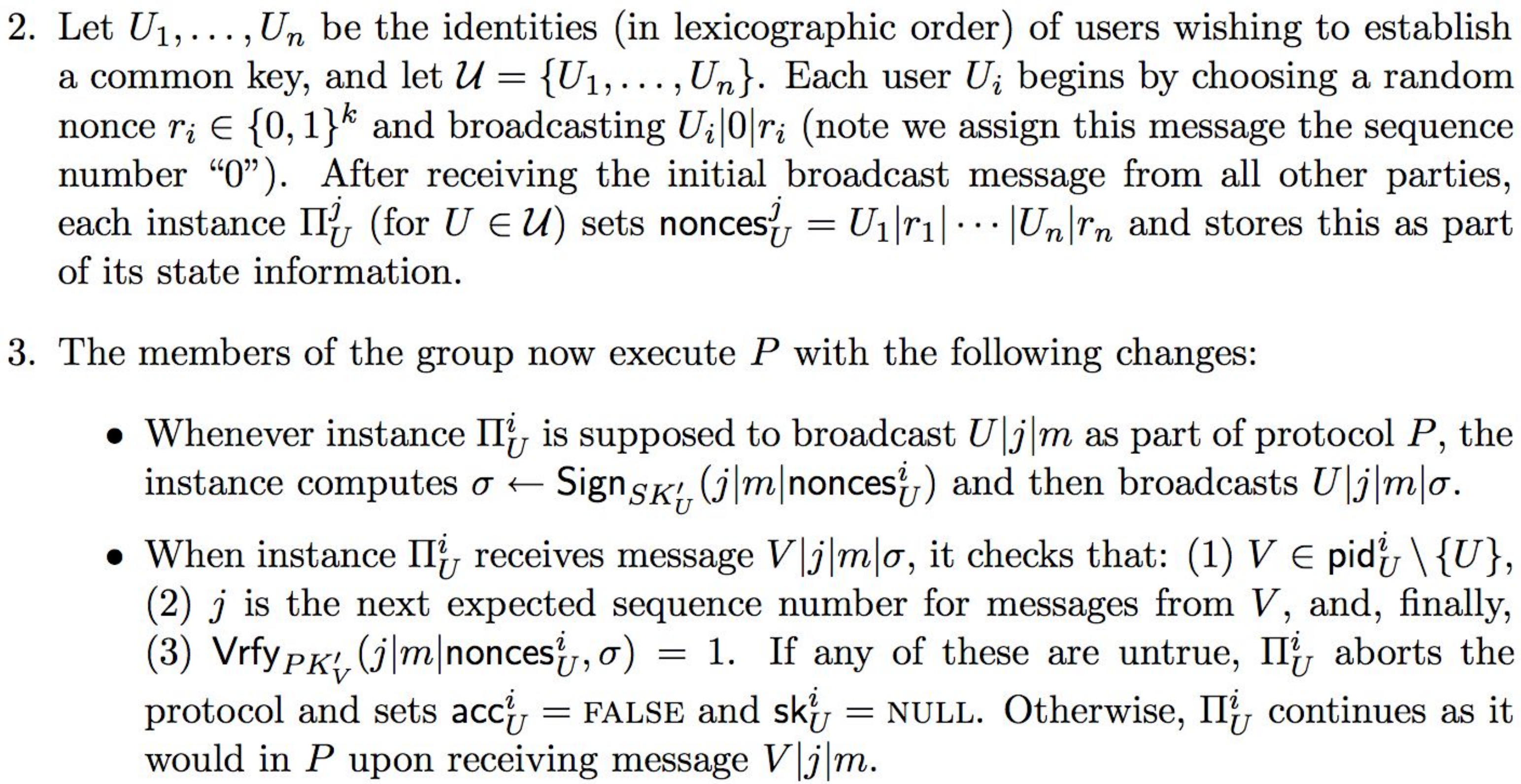
Approach



Cf. Sender Keys



Katz & Yung, 2013



2. Let $U_1, \ldots, U_n$ be the identities (in lexicographic order) of users wishing to establish a common key, and let $\mathcal{U} = \{U_1, \ldots, U_n\}$. Each user U_i begins by choosing a random nonce $r_i \in \{0,1\}^k$ and broadcasting $U_i|0|r_i$ (note we assign this message the sequence number "0"). After receiving the initial broadcast message from all other parties, each instance Π_U^j (for $U \in \mathcal{U}$) sets $\mathsf{nonces}_U^j = U_1|r_1|\cdots|U_n|r_n$ and stores this as part of its state information.

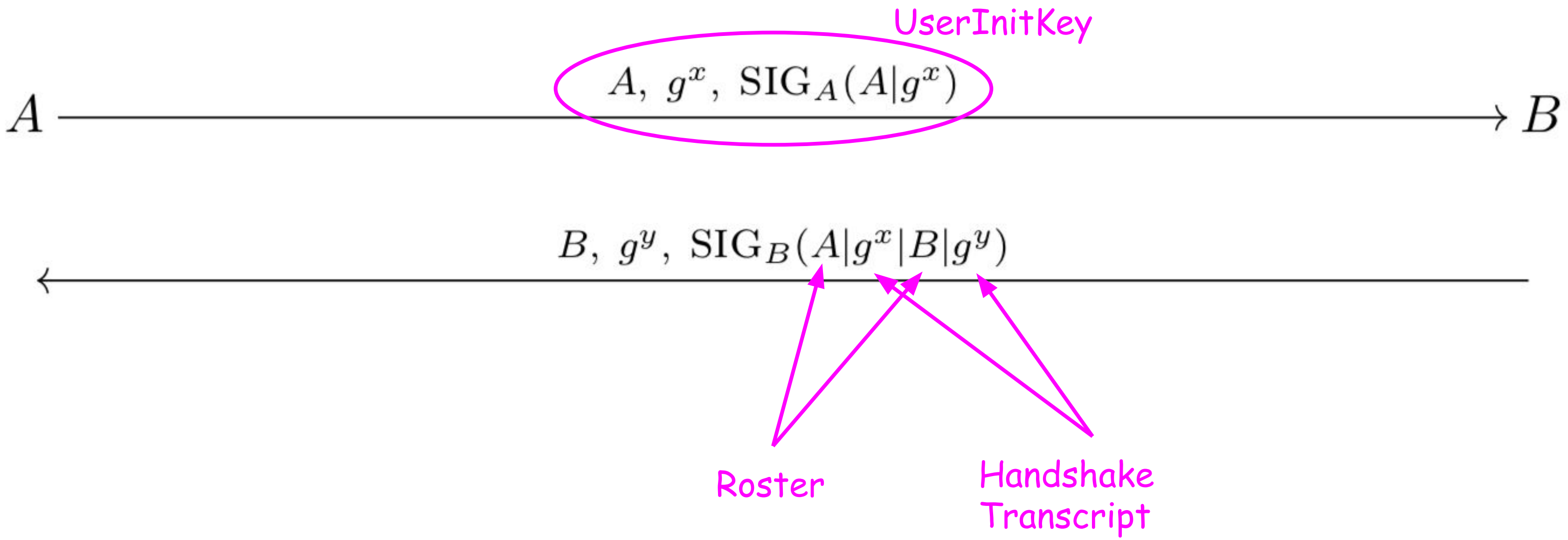
3. The members of the group now execute P with the following changes:

   - Whenever instance Π_U^i is supposed to broadcast $U|j|m$ as part of protocol P, the instance computes $\sigma \leftarrow \mathsf{Sign}_{SK'_U}(j|m|\mathsf{nonces}_U^i)$ and then broadcasts $U|j|m|\sigma$.
   - When instance Π_U^i receives message $V|j|m|\sigma$, it checks that: (1) $V \in \mathsf{pid}_U^i \setminus \{U\}$, (2) j is the next expected sequence number for messages from V, and, finally, (3) $\mathsf{Vrfy}_{PK'_V}(j|m|\mathsf{nonces}_U^i, \sigma) = 1$. If any of these are untrue, Π_U^i aborts the protocol and sets $\mathsf{acc}_U^i = \text{FALSE}$ and $\mathsf{sk}_U^i = \text{NULL}$. Otherwise, Π_U^i continues as it would in P upon receiving message $V|j|m$.

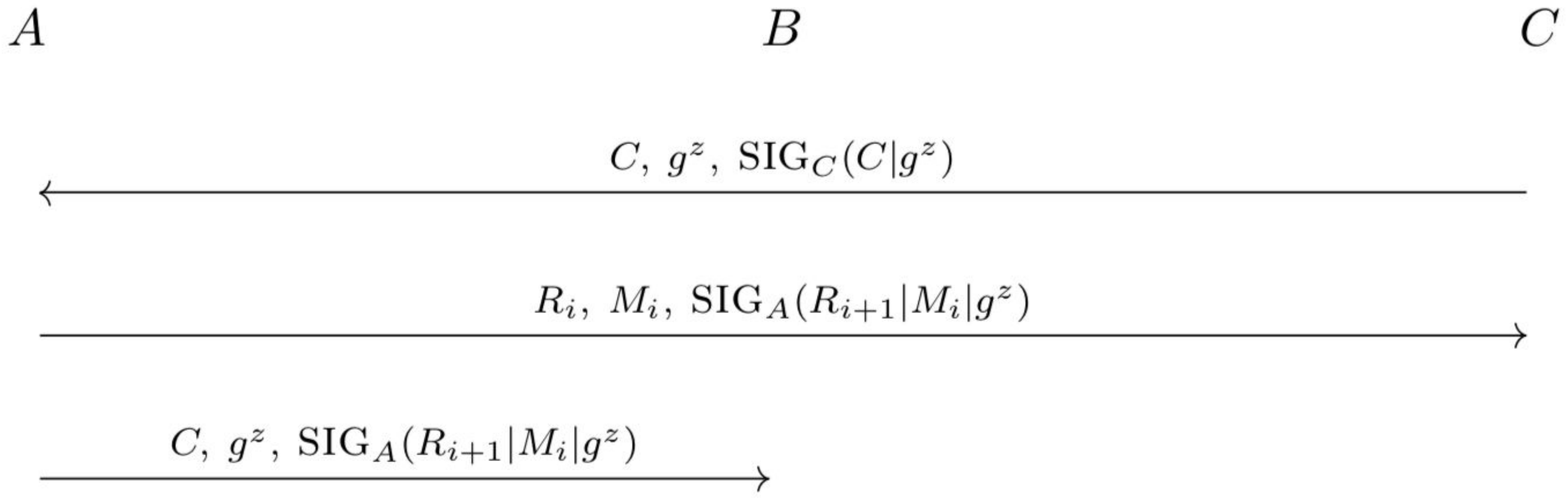
Katz-Yung, specialized to 2-party

$$\begin{tikzcd}
A \arrow[r, "{A,\ r_A}"] & B \\
{} & {} \arrow[l, "{B,\ r_B,\ g^y,\ \mathrm{SIG}_B(g^y|A|r_A|B|r_B)}"'] \\
{} \arrow[r, "{g^x,\ \mathrm{SIG}_A(g^x|A|r_A|B|r_B)}"] & {}
\end{tikzcd}$$

Katz-Yung, lightly rearranged


UserInitKey
$A,\ g^x,\ \mathrm{SIG}_A(A|g^x)$
A
B
$B,\ g^y,\ \mathrm{SIG}_B(A|g^x|B|g^y)$
Roster
Handshake Transcript

Katz-Yung, generalized back to groups


A
B
C
$C,\ g^z,\ \mathrm{SIG}_C(C|g^z)$
$R_i,\ M_i,\ \mathrm{SIG}_A(R_{i+1}|M_i|g^z)$
$C,\ g^z,\ \mathrm{SIG}_A(R_{i+1}|M_i|g^z)$

SIGMA vs. Katz-Yung

- Katz-Yung: $\textsc{sig}_A(R_{i+1}|M_i|g^z)$
- SIGMA: $\textsc{sig}_A(M_i|g^z),\ \textsc{mac}_{K_i}(R_{i+1})$

Possible to extend SIGMA in a similar way

Pro Katz-Yung: Slightly simpler (just a signature)

Pro SIGMA: Demonstrates that sender holds group key (in addition to being an authorized signer)

Is it Secure?

Tamarin models seem to indicate so:

- 3-4 party implementations verified to match same properties as 2-party
- … with both Katz-Yung-like and SIGMA-like flavors
- https://github.com/bifurcation/tamarin-ake/

Would appreciate validation that:

- Verified properties are the correct ones
- Proofs generalize to all group sizes

As implemented

Group members keep a GroupState object that reflects the things the group should agree on (except secrets)

GroupState gets:

- Hashed into keys
- Signed by message senders

```
struct {
    opaque group_id<0..255>;
    uint32 epoch;
    Credential roster<1..2^32-1>;
    PublicKey tree<1..2^32-1>;
    GroupOperation transcript<0..2^32-1>;
} GroupState;
```

```
Derive-Secret(Secret, Label, State, Length)
```

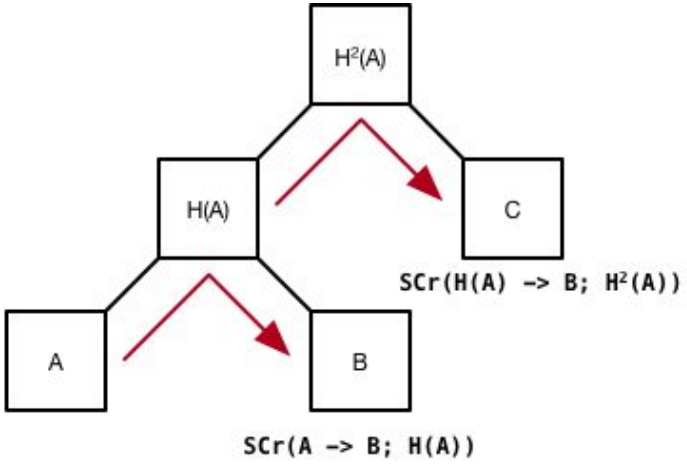
```
struct {
    uint32 prior_epoch;
    GroupOperation operation;

    uint32 signer_index;
    SignatureScheme algorithm;
    opaque signature<1..2^16-1>;
} Handshake;
```

Aside: “Signcryption”

NaCl “box” operation does “signcryption”: Static DH keys for both sender and receiver are mixed into encryption key

This could be useful for authenticating Update messages, to authenticate possession of relevant tree keys

Would cost an extra DH operation on receiver; no message overhead



Questions

Does anyone see problems with the proposed GroupState-based scheme?

Preferences for Katz-Yung vs. SIGMA?

Are people OK with caching the whole roster / tree?

Does the idea of incorporating signcryption seem worthwhile?