Cryptographic Administration for Secure Group Messaging

David Balbás[*†], Daniel Collins[‡], Serge Vaudenay[‡]

[*]IMDEA Software Institute, Madrid, Spain
[†]Universidad Politécnica de Madrid, Spain
[‡]EPFL, Lausanne, Switzerland

9th August 2023

USENIX Security '23, Anaheim, CA

thenewsminute.com

WhatsApp Group chats can be easily infiltrated, say researchers

Written by IANS

4–5 minutes

The WhatsApp attack on group chats takes advantage of a bug.

A team of German cryptographers has discovered flaws in WhatsApp's Group chats despite its end-to-end encryption, that makes it possible to infiltrate private group chats without admin permission.

According to a report in Wired.com, the cryptographers from Ruhr University Bochum in Germany announced this at the "Real World Crypto Security Conference in Zurich, Switzerland, on Wednesday.

"Anyone who controls the app's servers could insert new people into private group chats without needing admin permission," the report said, citing cryptographers.

thenewsminute.com

WhatsApp Group chats can be easily infiltrated, say researchers

Written by IANS

4–5 minutes

The WhatsApp attack on group

A team of German cryptograph
WhatsApp's Group chats desp
makes it possible to infiltrate p
permission.

According to a report in Wired.
University Bochum in Germany
Crypto Security Conference in

"Anyone who controls the app'
into private group chats withou
report said, citing cryptographe

ISG researchers discover vulnerabilities in Matrix protocol

Research and teaching › Departments and schools › Information Security › News

Date **28 September 2022**

A team of cryptographers – Dan Jones and Martin Albrecht (Royal Holloway), Sofía Celi (Brave) and Benjamin Dowling (University of Sheffield) has found several, practically-exploitable cryptographic vulnerabilities in the end-to-end encryption provided by the popular Matrix protocol and its flagship client implementation Element.

thenewsminute.com

WhatsApp Group chats can be easily infiltrated, say researchers

Written by IANS

4–5 minutes

The WhatsApp attack on group

A team of German cryptograph

ISG researchers discover vulnerabilities in Matrix protocol

Research and teaching › Departments and schools › Information Security › News

Date 28 September 2022

Three Lessons From Threema: Analysis of a Secure Messenger

Kenneth G. Paterson
Applied Cryptography Group, ETH Zurich

Matteo Scarlata
Applied Cryptography Group, ETH Zurich

Kien Tuong Truong
Applied Cryptography Group, ETH Zurich

Abstract

- **fine-grained perfect forward secrecy (PFS)**: compro-

echt
ng

end
d its

Why Administration?

Insecure group membership is a common design flaw in messaging.

Servers, and sometimes even users, may mount **attacks on group management**.

- Burgle into the group [RMS18]
- Censorship [BCG23]
- ...

Why Administration?

Insecure group membership is a common design flaw in messaging.

Servers, and sometimes even users, may mount **attacks on group management**.

- Burgle into the group [RMS18]
- Censorship [BCG23]
- ...

How meaningful is security if users can't trust/control group membership?

Why Administration?

Insecure group membership is a common design flaw in messaging.

Servers, and sometimes even users, may mount **attacks on group management**.

- Burgle into the group [RMS18]
- Censorship [BCG23]
- ...

How meaningful is security if users can't trust/control group membership?

Can we build an efficient solution for users to *administrate* groups securely?

This Work

- New **formalism** for groups with *cryptographic administrators*.

This Work

- New **formalism** for groups with *cryptographic administrators*.
- **Correctness and security notions** matching modern messaging standards (forward security, post-compromise security).

This Work

- New **formalism** for groups with *cryptographic administrators*.
- **Correctness and security notions** matching modern messaging standards (forward security, post-compromise security).
- Two modular, **provably-secure constructions**, IAS and DGS.

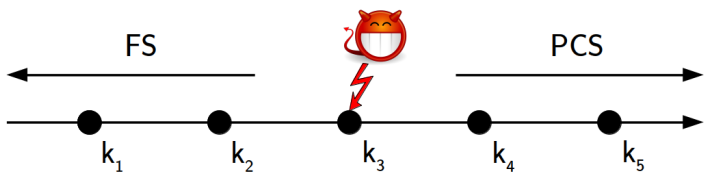
This Work

- New **formalism** for groups with *cryptographic administrators*.
- **Correctness and security notions** matching modern messaging standards (forward security, post-compromise security).
- Two modular, **provably-secure constructions**, IAS and DGS.
- Efficient **integration with MLS**, admin extensions.

Group Messaging

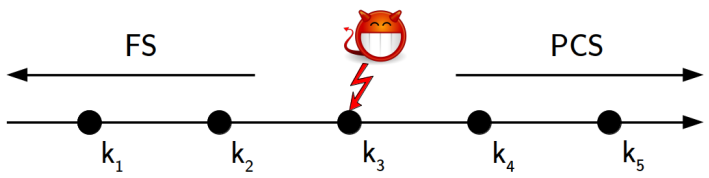
Security of Group Messaging

- **Forward security (FS):** past messages safe after compromise.
- **Post-compromise security (PCS):** self-healing via key updates.



Security of Group Messaging

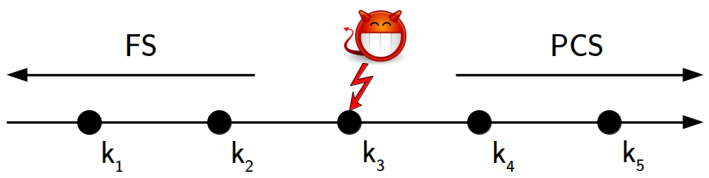
- **Forward security (FS):** past messages safe after compromise.
- **Post-compromise security (PCS):** self-healing via key updates.



- **Security game:** $\mathcal{A}$ controls network, can expose users [ACDT20, KPWK+21].
- **Group dynamics:** cryptographic adds and removes from group G.

Security of Group Messaging

- **Forward security (FS):** past messages safe after compromise.
- **Post-compromise security (PCS):** self-healing via key updates.



- **Security game:** $\mathcal{A}$ controls network, can expose users [ACDT20, KPWK+21].
- **Group dynamics:** cryptographic adds and removes from group G.
- ***Administration:*** only admins $G^* \subseteq G$ can make group changes.

Key Agreement: (A-)CGKA

Popular formalism: **Continuous Group Key Agreement** (CGKA) [ACDT20]. Basis of MLS.

Key Agreement: (A-)CGKA

Popular formalism: **Continuous Group Key Agreement** (CGKA) [ACDT20]. Basis of MLS.

- Dynamic *secret* / known to members.

Key Agreement: (A-)CGKA

Popular formalism: **Continuous Group Key Agreement** (CGKA) [ACDT20]. Basis of MLS.

- Dynamic *secret* I known to members.
- Members *ID propose* adds, removals, and key updates [AJM20, RFC9420].

Key Agreement: (A-)CGKA

Popular formalism: **Continuous Group Key Agreement** (CGKA) [ACDT20]. Basis of MLS.

- Dynamic *secret* I known to members.
- Members ID *propose* adds, removals, and key updates [AJM20, RFC9420].
- Later, ID' *commits* several proposals.

Key Agreement: (A-)CGKA

Popular formalism: **Continuous Group Key Agreement** (CGKA) [ACDT20]. Basis of MLS.

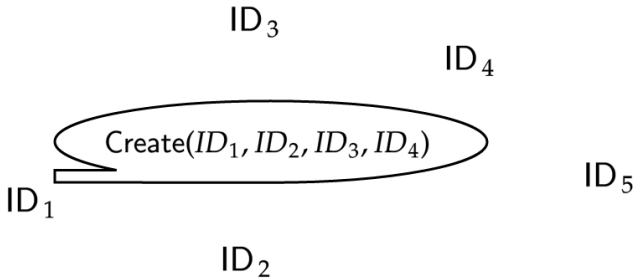
- Dynamic *secret* I known to members.
- Members ID *propose* adds, removals, and key updates [AJM20, RFC9420].
- Later, ID' *commits* several proposals.

CGKA (simpl.):

- $\mathsf{Init}(1^\lambda, ID)$
- $\mathsf{Create}(G) \to T$
- $\mathsf{Prop}(ID, \mathsf{type}) \to P$
- $\mathsf{Commit}(\vec{P}) \to T$
- $\mathsf{Proc}(T) \to I'$

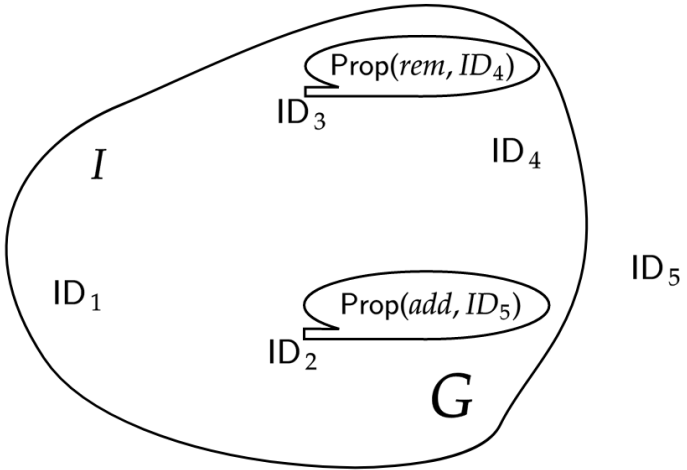
CGKA: Create

- ID_1 creates a group $G = \{ID_1, ID_2, ID_3, ID_4\}$.



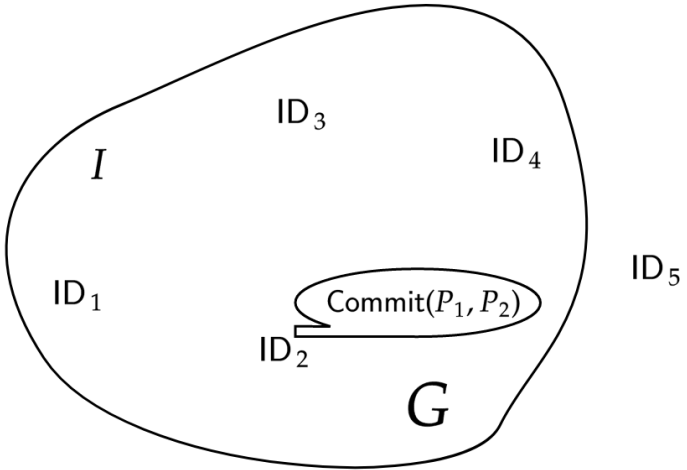
CGKA: Proposals

- ID_2 and ID_3 propose changes.



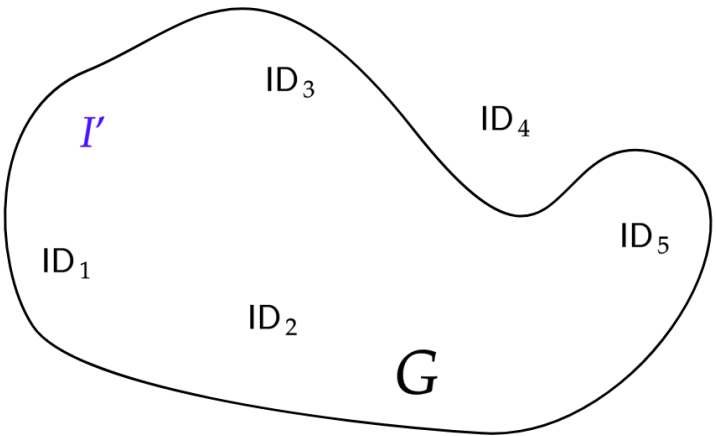
CGKA: Commit

- ID_2 commits both proposals.



CGKA: Process Changes

- The group evolves and I' is refreshed.



Key Agreement: (A-)CGKA

Administrated Continuous Group Key Agreement (A-CGKA).

- Dynamic *secret* I known to members.
- Members ID *propose* adds, removals, and key updates [AJM20, RFC9420].

  A-CGKA includes new proposal types: add/remove/update admin.
- Later, ID' *commits* several proposals.

A-CGKA (simpl.):

- $\text{Init}(1^\lambda, ID)$
- $\text{Create}(G, G^*) \to T$
- $\text{Prop}(ID, \text{type}) \to P$
- $\text{Commit}(\vec{P}, \text{com-type}) \to T$
- $\text{Proc}(T) \to I'$

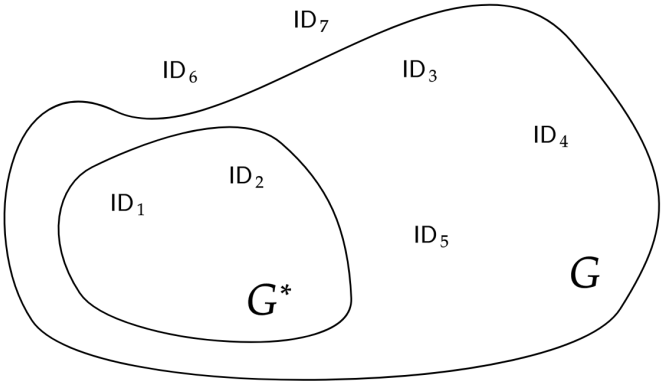
Administration security: Non-admins cannot commit (except updates and self-removes).

Protocols for Secure Administration

We introduce **IAS** (*Individual Admin Signatures*) and **DGS** (*Dynamic Group Signature*).

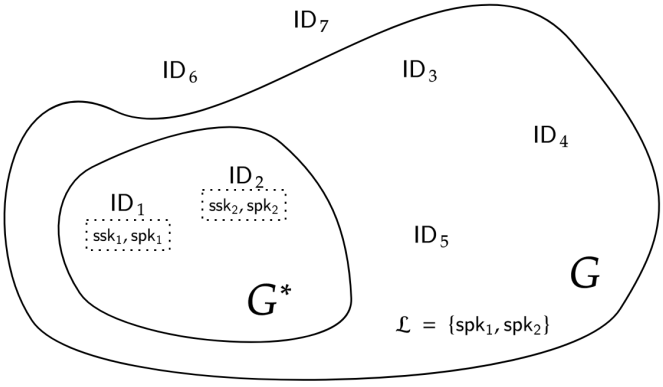
- Modular.
- Authenticate administrators (with different efficiency trade-offs).
- Allow for admin key refresh for PCS and FS.

Individual Admin Signatures (IAS)



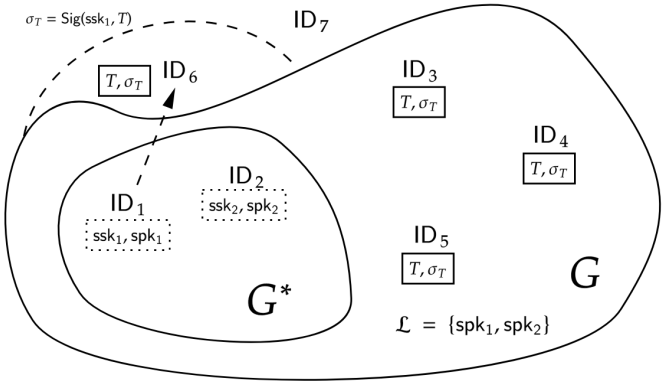
- We construct A-CGKA on top of any CGKA.
- Based on signatures.

IAS



- Admins have individual signature key pairs $(\mathsf{ssk}, \mathsf{spk})$.
- Users keep an admin list $\mathcal{L}$.

IAS: Add Participant



- Admin signs commit T with $\mathsf{ssk}_1 \longrightarrow \sigma_T$.
- Users verify σ_T with spk_1 from $\mathcal{L}$.

Practical Administration for MLS

We also **integrate an IAS-based solution into MLS:**

Practical Administration for MLS

We also **integrate an IAS-based solution into MLS:**

- Updates for MLS' key credentials.
- Extended proposal types.

Practical Administration for MLS

We also **integrate an IAS-based solution into MLS:**

- Updates for MLS' key credentials.
- Extended proposal types.
- **Minimal overhead** (from benchmarking):
  - *Less than 20%* when $|G|/8$ members update simultaneously.
  - *Additional communication* $< 3\%$ for $|G| = 128$ members.

Conclusions

- Securing *membership* is essential in group messaging security.
- Cryptographic *administration* can be implemented with small overhead.
- Modular solutions *readily compatible* with CGKAs and MLS.

Conclusions

- Securing *membership* is essential in group messaging security.
- Cryptographic *administration* can be implemented with small overhead.
- Modular solutions *readily compatible* with CGKAs and MLS.

Open Problems:

- Prevent insider attacks efficiently.
- Advanced admin functionalities.
- Admins beyond CGKA.

Conclusions

- Securing *membership* is essential in group messaging security.
- Cryptographic *administration* can be implemented with small overhead.
- Modular solutions *readily compatible* with CGKAs and MLS.

Open Problems:

- Prevent insider attacks efficiently.
- Advanced admin functionalities.
- Admins beyond CGKA.

Thank you!

ia.cr/2022/1411

david.balbas@imdea.org
daniel.collins@epfl.ch

Join us at the **poster session** to find out about Sender Keys security!