On the Cost of Post-Compromise Security in Concurrent Continuous Group-Key Agreement

Benedikt Auerbach, Miguel Cueto Noval, Guillermo Pascual-Perez, Krzysztof Pietrzak

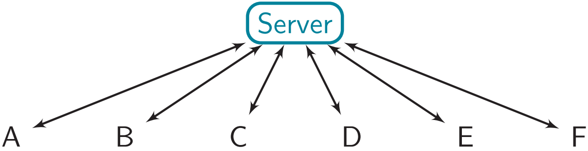
ISTA, Austria

TCC 2023

Institute of Science and Technology Austria

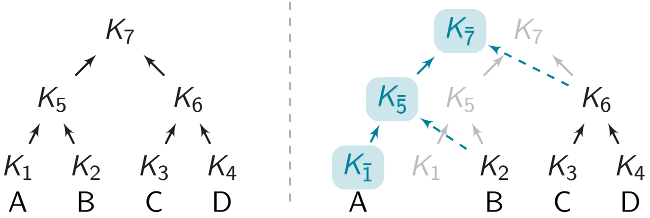
Group Messaging

- Messaging apps: WhatsApp, Signal...
- Asynchronous and long lived sessions.



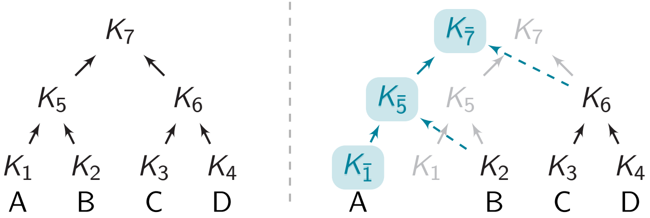
Continuous Group Key Agreement (CGKA)

- A CGKA is a protocol that allows a group of users to maintain a shared key and update it.



Continuous Group Key Agreement (CGKA)

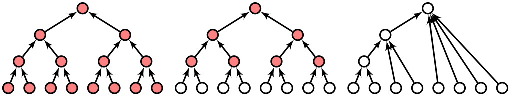
- A CGKA is a protocol that allows a group of users to maintain a shared key and update it.
- Users update the key material to achieve Post-Compromise Security (PCS).
- Protocols: ART [CCG+18], TreeKEM [BBR18], Causal TreeKEM [Mat19], rTreeKEM [ACDT20], Tainted TreeKEM [KPPW+21], decentralized CGKA [WKHB21], CoCoA [AAN+22a], DeCAF [AAN+22b].



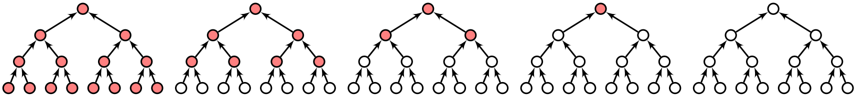
Concurrent Updates

What happens when several users want to do an update in round t?

Propose and Commit



CoCoA

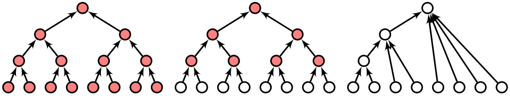


Source: CoCoA [AAN+22a]

Concurrent Updates

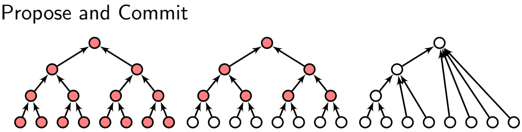
- What happens when several users want to do an update in round t?
- Propose and Commit (P&C) makes future updates inefficient. Worst-case total upload cost n^2.

Propose and Commit



Concurrent Updates

- What happens when several users want to do an update in round t?
- Propose and Commit (P&C) makes future updates inefficient. Worst-case total upload cost n^2.
- Worst-case lower bound for 2-PCS [BDR20] of n^2 (in a symbolic model).



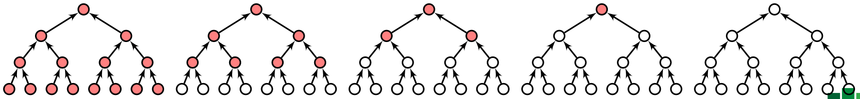
Concurrent Updates

- What happens when several users want to do an update in round t?
- For 2-PCS, optimal cost is n^2.

Concurrent Updates

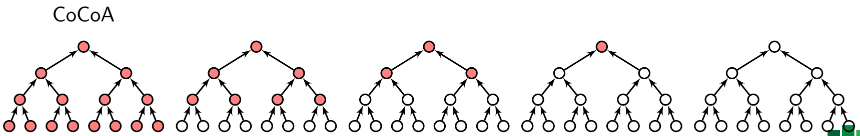
- What happens when several users want to do an update in round t?
- For 2-PCS, optimal cost is n^2.
- How about delaying PCS? CoCoA [AAN+22a] achieves PCS in $\log(n)$ rounds with total upload cost $n \log^2 n$.

CoCoA

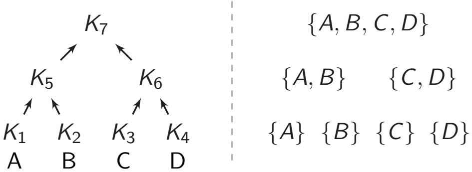


Concurrent Updates

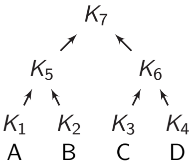
- What happens when several users want to do an update in round t?
- For 2-PCS, optimal cost is n^2.
- How about delaying PCS? CoCoA [AAN+22a] achieves PCS in $\log(n)$ rounds with total upload cost $n \log^2 n$.
- **This work: Lower Bound for concurrency in k-PCS and almost matching upper bound. Applies to a large class of protocols.**



The Combinatorial Model in Pictures



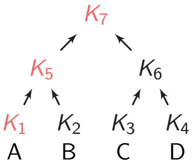
The Combinatorial Model in Pictures



$\{A, B, C, D\}$

$\{A, B\}$ $\{C, D\}$

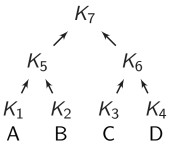
$\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$



$\{C, D\}$

$\{B\}$ $\{C\}$ $\{D\}$

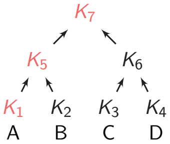
The Combinatorial Model in Pictures



$\{A, B, C, D\}$

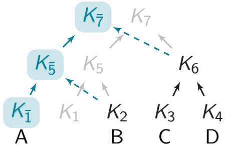
$\{A, B\}$ $\{C, D\}$

$\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$



$\{C, D\}$

$\{B\}$ $\{C\}$ $\{D\}$



$\{A, B, C, D\}$

$\{A, B\}$ $\{C, D\}$

$\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$

$\{A, B, C, D\} = \{A\} \cup \{B\} \cup \{C, D\}$

$\mathrm{Cost}(\{A, B, C, D\}) = 2$

Lower Bound in the Combinatorial Model

Lower Bound in the Combinatorial Model

Theorem (Informal Statement)

Let $t_c \in \mathbb{N}$ and $0 < \varepsilon < 2/5$ be a constant such that $(1+\varepsilon)k \in \mathbb{N}$. For every sequence of updates $(U_t)_{t=1}^{t_c}$, there exist a choice of C_{t_c} and a sequence of updates $(U_{t_c+t})_{t=1}^{(1+\varepsilon)k}$ such that

$$\sum_{t=1}^{(1+\varepsilon)k} \mathrm{Cost}(U_{t_c+t}) = \Omega(k \cdot n^{1+1/(k-1)} / \log(k)) \ .$$

And the same lower bound holds in the symbolic model for any correct and k-PCS secure CGKA protocol.

Upper Bounds

- CoCoALight and CoCoALight+ based on CoCoA [AAN+22a].

Lower Bounds		
[BDR20]	$k = 2$	$\Omega(n^2)$
This work	$k = 2$	$\Omega(n^2)$
	$k = \log n$	$\Omega(n \log n / \log \log n)$
	k	$\Omega(k \cdot n^{1+1/(k-1)} / \log(k))$
Upper Bounds		
P&C	$k = 2$	$O(n^2)$
CoCoA [AAN+22a]	$k = \log n$	$O(n \log^2 n)$
This work: CoCoALight	$k = \log n$	$O(n \log n)$