Fully Secure Broadcast Encryption

D. Hieu Phan, David Pointcheval, *Mario Strefler*

INRIA/ENS
strefler@di.ens.fr

2011 April 08

Broadcast Encryption

- N users $\{u_1, \ldots, u_N\} = U$
- Here: Key encapsulation mechanism
- Goal: Encrypt K to any $S \subset U$

Broadcast Encryption

- N users $\{u_1, \ldots, u_N\} = U$
- Here: Key encapsulation mechanism
- Goal: Encrypt K to any $S \subset U$
- Naive way: Use public-key encryption
- but: $|c| = \mathcal{O}(|S|)$

Subset Cover

- Partition user set into subsets S_i
- Assign each subset a key
- Each user gets keys of sets he belongs to

Subset Cover

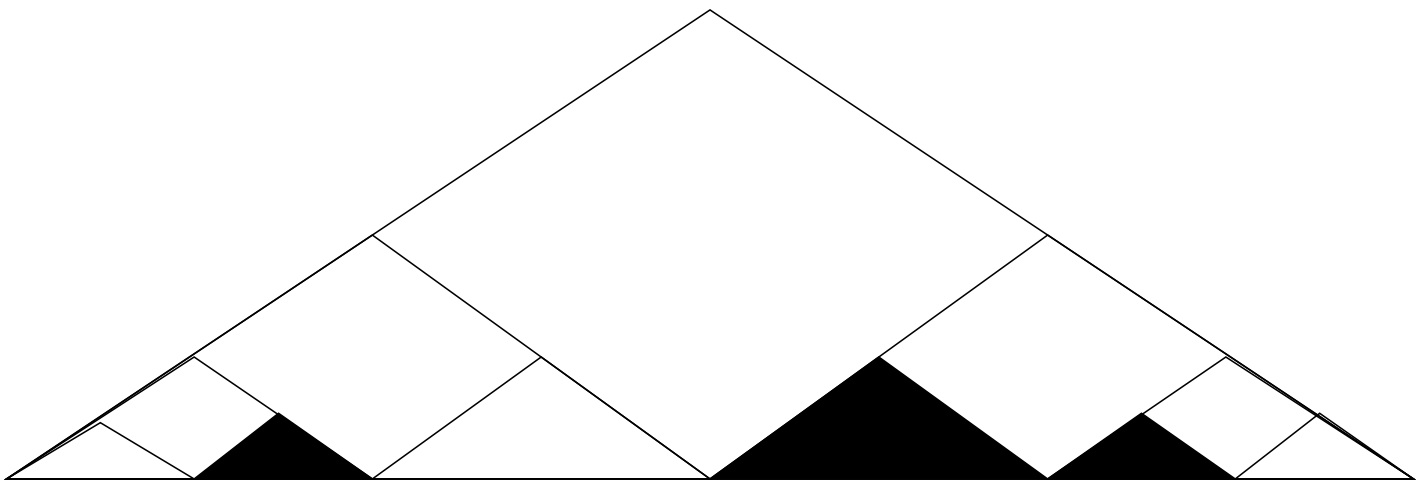
- Partition user set into subsets S_i
- Assign each subset a key
- Each user gets keys of sets he belongs to

To encrypt

- Partition target set into subsets S_i
- Use subset keys to encaps session key

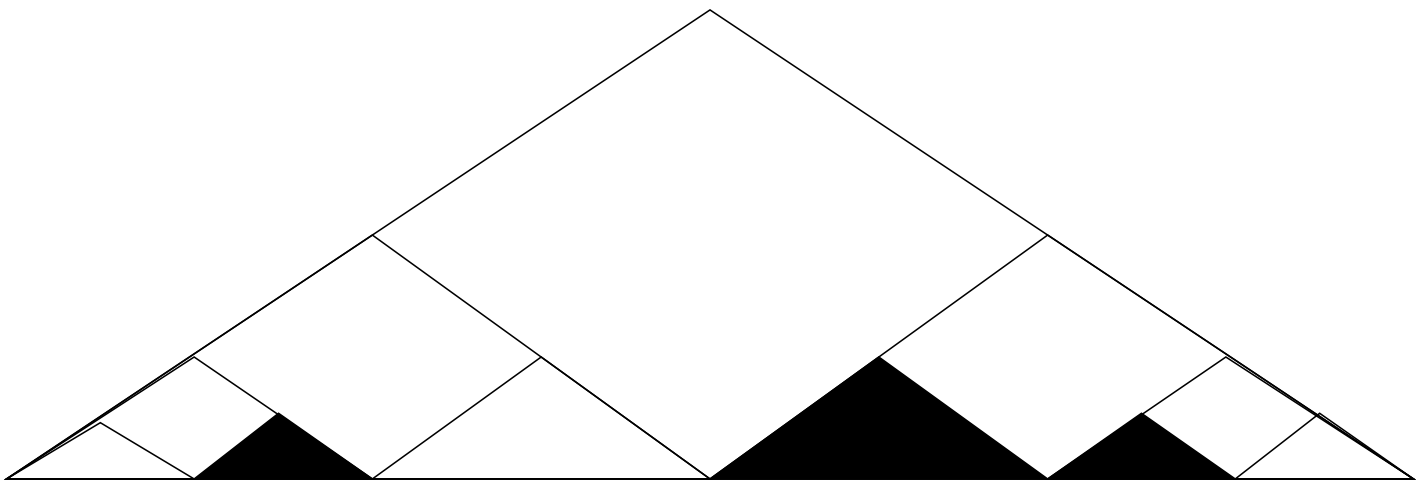
Complete Subtree

- Users are leafs in a tree
- Each node i is assigned a key for subtree S_i
- Each users has keys of all ancestors $(\log N)$



Complete Subtree

- Users are leafs in a tree
- Each node i is assigned a key for subtree S_i
- Each users has keys of all ancestors $(\log N)$



- $|c| = \mathcal{O}(r \log N)$
- size of the user secret: $\log N$

Tools

Diffie-Hellman Key Agreement

- u_a has secret key a, u_b has b
- public: $A = g^a, B = g^b$
- Compute session key $A^b = g^{ab} = B^a$

Tools

Diffie-Hellman Key Agreement

- u_a has secret key a, u_b has b
- public: $A = g^a, B = g^b$
- Compute session key $A^b = g^{ab} = B^a$

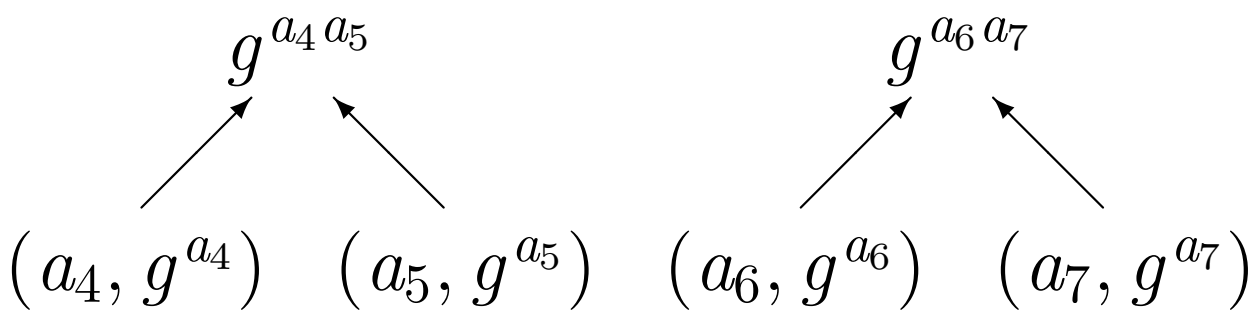
ElGamal Encryption

- Receiver has $dk = a$, $ek = A = g^a$
- To encrypt m choose b and send $g^b, A^b m$
- IND-CPA secure under DDH

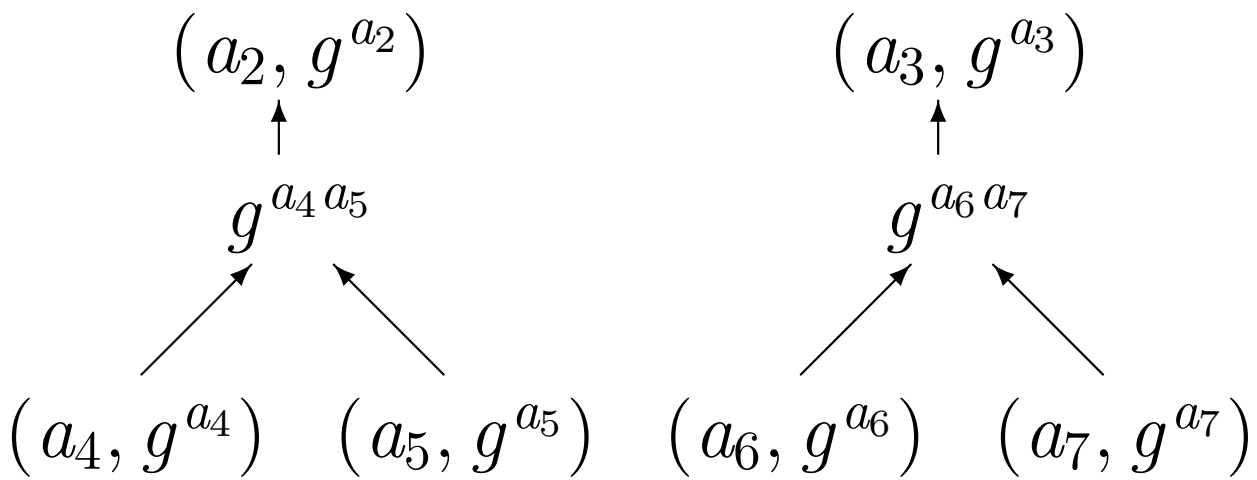
Tree Construction

$$(a_4, g^{a_4}) \quad (a_5, g^{a_5}) \quad (a_6, g^{a_6}) \quad (a_7, g^{a_7})$$

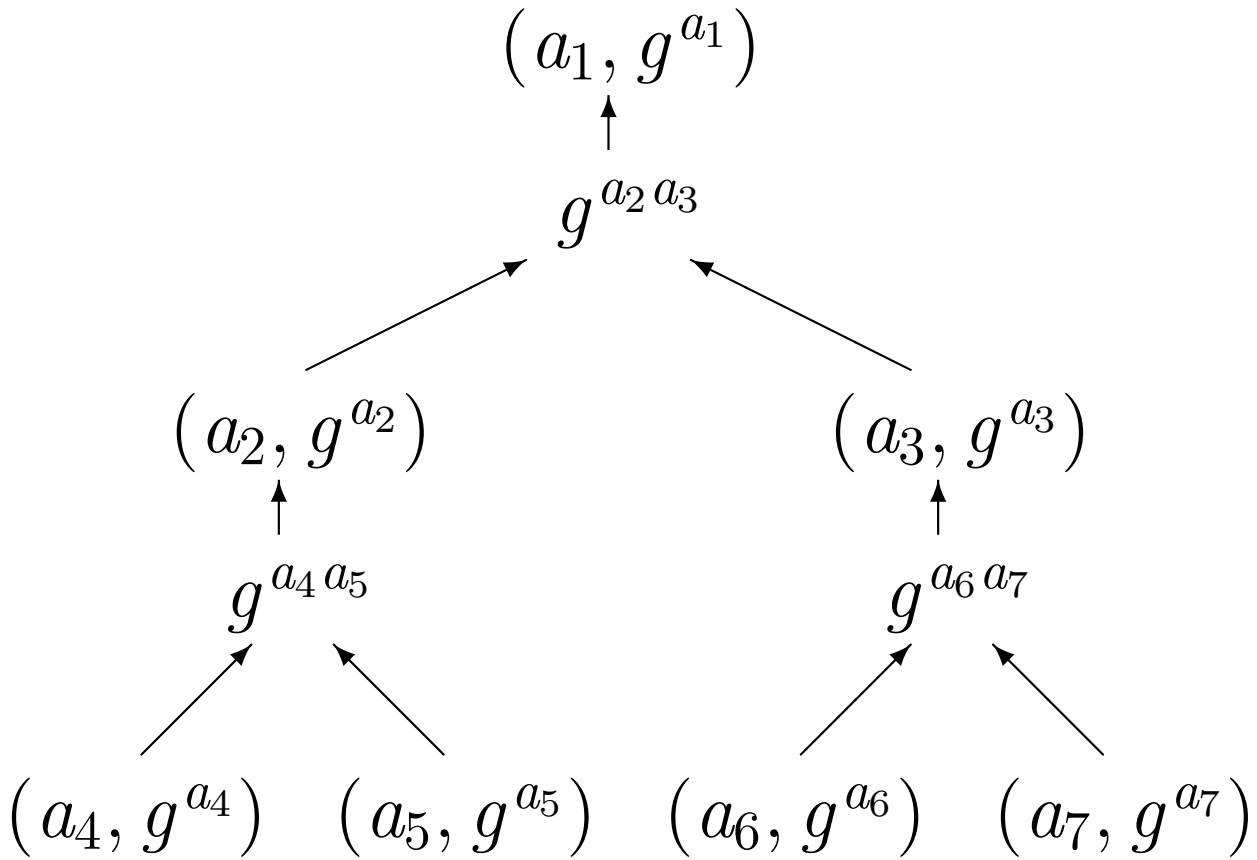
Tree Construction



Tree Construction



Tree Construction

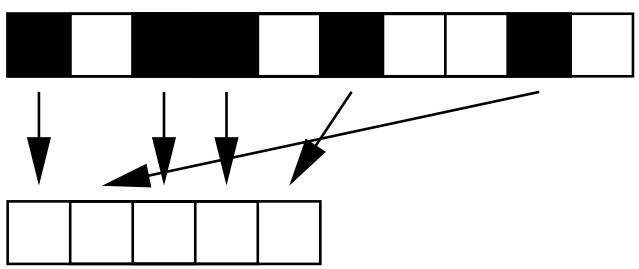


Can reconstruct tree from constant-size secret

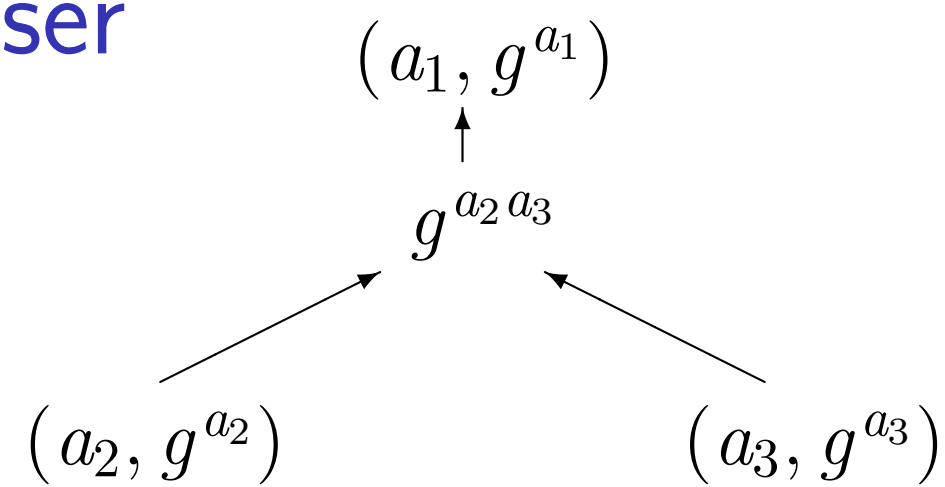
The PRG

Want to go from a subgroup of $\mathbb{Z}_p^\times$ to $\mathbb{Z}_q$ $(p = 2q + 1)$

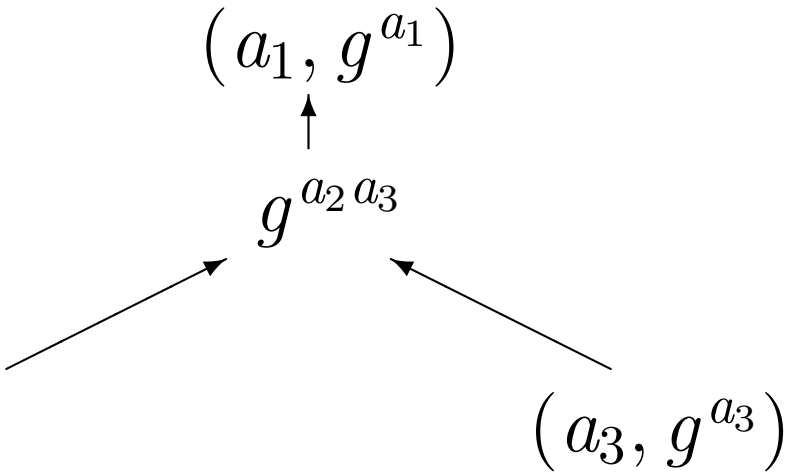
$$f(x) = \begin{cases} x & \texttt{if } x \leq q \\ p - x & \texttt{if } x > q \end{cases}$$



Adding a user

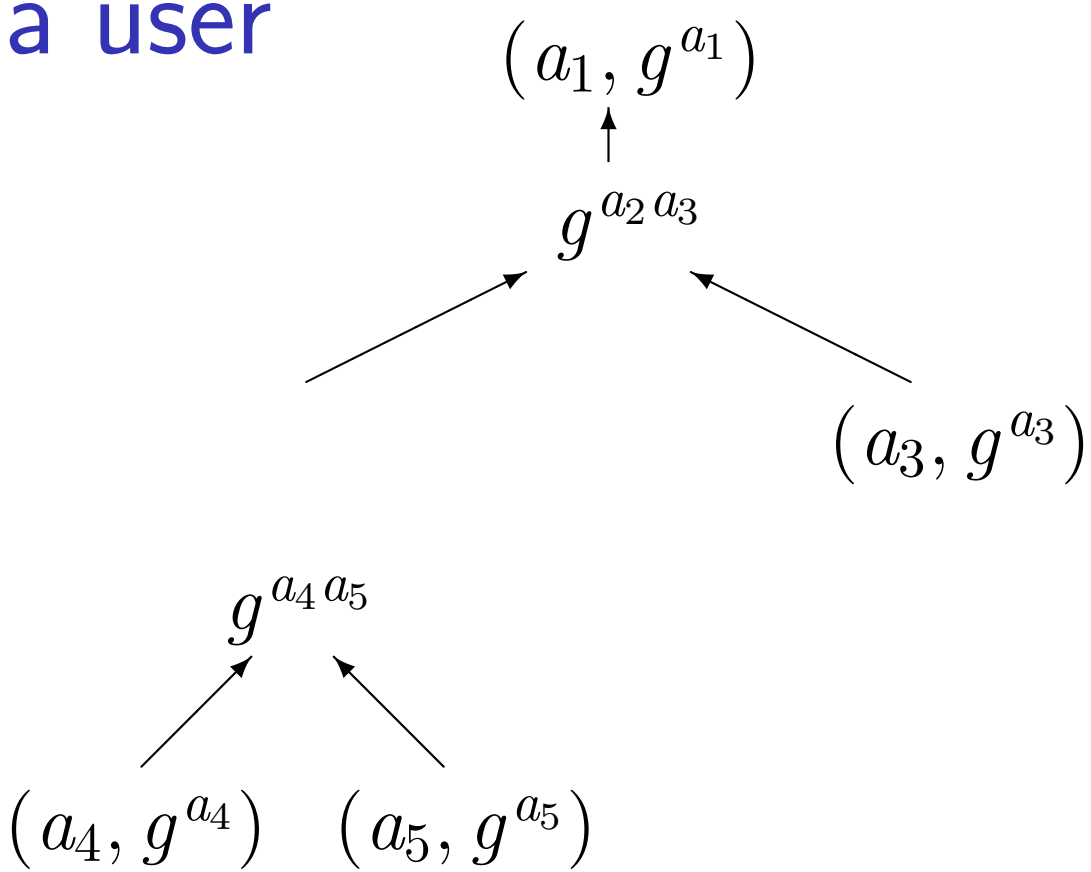


Adding a user

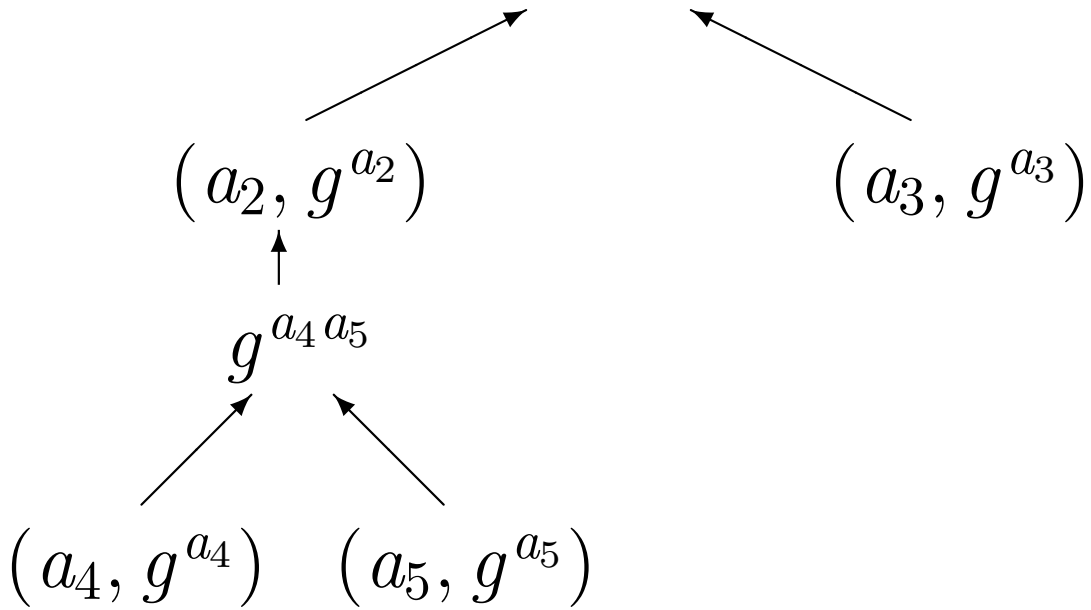


(a_4, g^{a_4}) (a_5, g^{a_5})

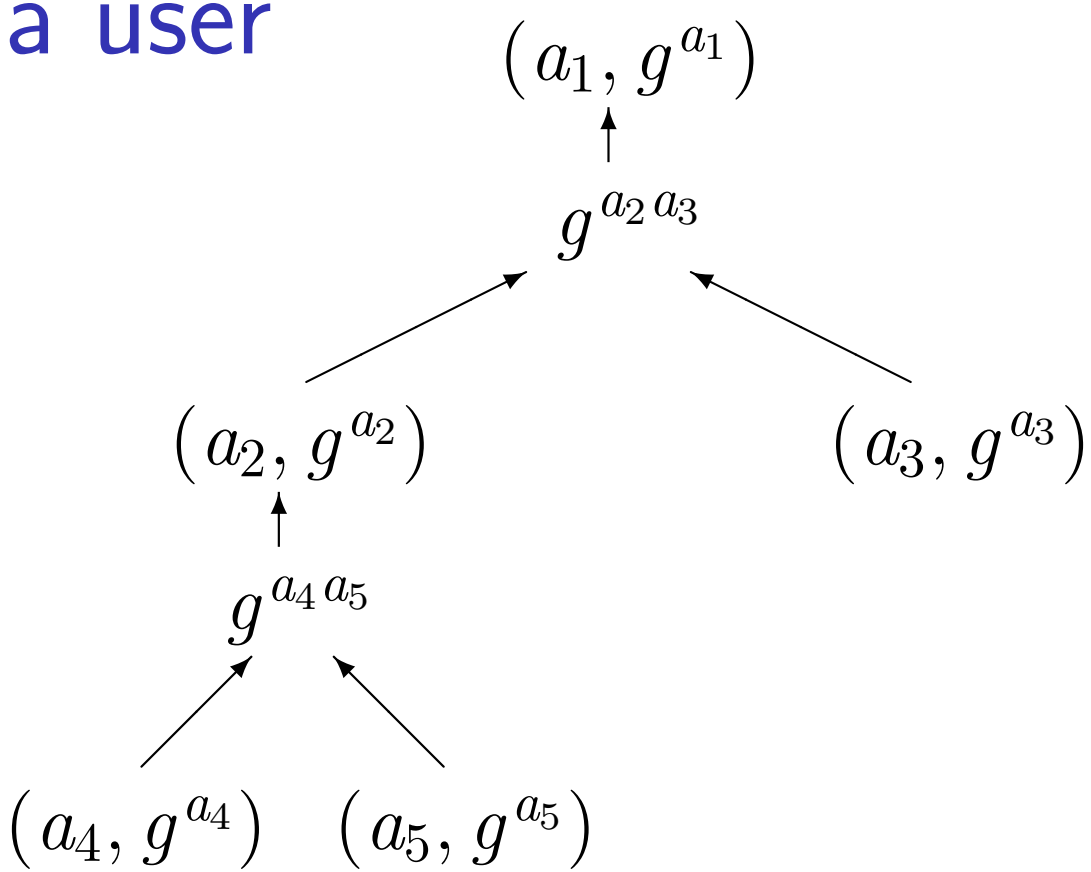
Adding a user



Adding a user



Adding a user



Need to recompute all key pairs on the path to the root.

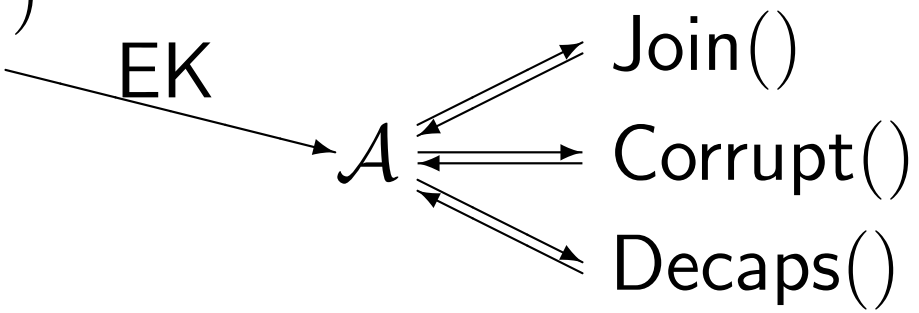
Security of BE

- Have: IND-CPA with fully adaptive corruption
- Want: IND-CCA2 (also fully adaptive)

Security of BE

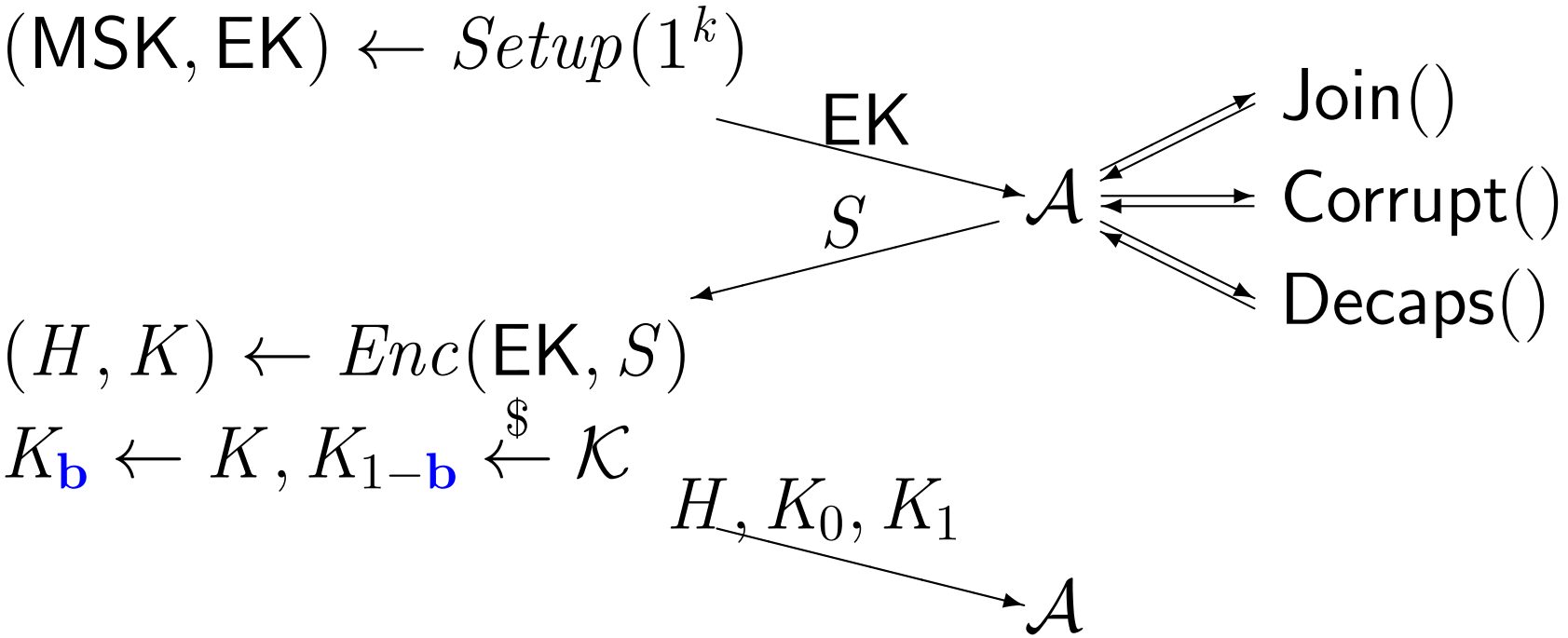
- Have: IND-CPA with fully adaptive corruption
- Want: IND-CCA2 (also fully adaptive)

$(\mathsf{MSK}, \mathsf{EK}) \leftarrow Setup(1^k)$



Security of BE

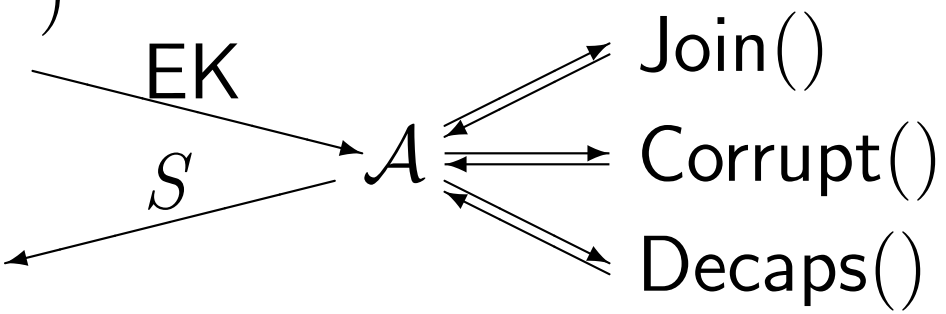
- Have: IND-CPA with fully adaptive corruption
- Want: IND-CCA2 (also fully adaptive)



Security of BE

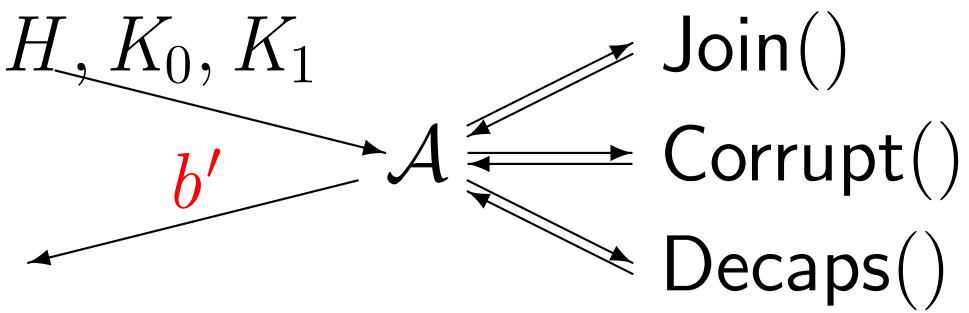
- Have: IND-CPA with fully adaptive corruption
- Want: IND-CCA2 (also fully adaptive)

$(\mathsf{MSK}, \mathsf{EK}) \leftarrow Setup(1^k)$



$(H, K) \leftarrow Enc(\mathsf{EK}, S)$

$K_{\mathbf{b}} \leftarrow K, K_{1-\mathbf{b}} \xleftarrow{\$} \mathcal{K}$



Cramer-Shoup Encryption

- IND-CCA2 under DDH
- $dk = (x_1, x_2, y_1, y_2, z)$, $ek = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{x_2}, g_1^z)$
- Also need MAC to tie components together

Generation of Pseudorandomness

- We can construct a KA tree
- Need to also generate PKE keys (5 el.)

Generation of Pseudorandomness

- We can construct a KA tree
- Need to also generate PKE keys (5 el.)
- Multi-DH: $sk_i = (a_i, b_i, c_i)$;
  $pk_i = (A_i, B_i, C_i) = (g^{a_i}, g^{b_i}, g^{c_i})$.
- The session key is: (3+5 el.)

	a_j	b_j	c_j
A_i	$A_i^{a_j}$	$A_i^{b_j}$	$A_i^{c_j}$
B_i	$B_i^{a_j}$	$B_i^{b_j}$	$B_i^{c_j}$
C_i	$C_i^{a_j}$	$C_i^{b_j}$	

Outlook

- Subset Difference: $|c| = \mathcal{O}(r)$
- Permanently revoke users
- Remove central authority by making the key agreement interactive

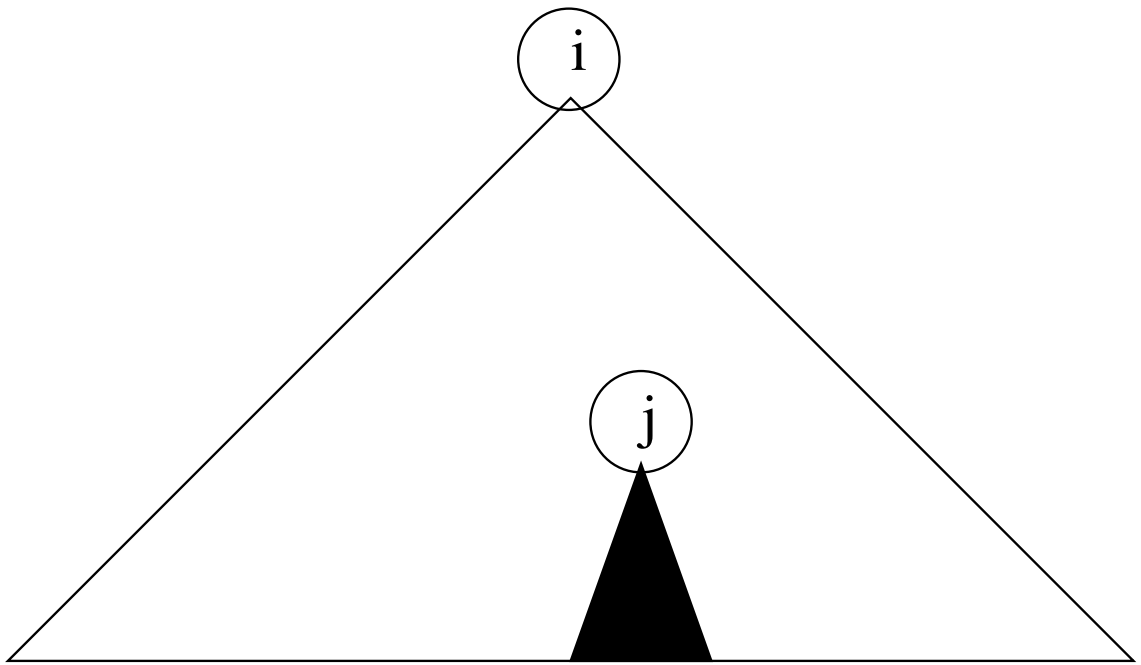
Thank you.

Subset Difference

- Users are leafs in a tree
- Subset $S_{i,j} = S_i \setminus S_j$
- Each users has $\log^2 N$ keys
- $|c| = \mathcal{O}(r)$

Subset Difference

- Users are leafs in a tree
- Subset $S_{i,j} = S_i \setminus S_j$
- Each users has $\log^2 N$ keys
- $|c| = \mathcal{O}(r)$



Security of BE

$\mathsf{Exp}^{\mathsf{ind-acca}-b}_{\mathcal{DBE},\mathcal{A}}(k)$
$(\mathsf{MSK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^k)$;
$\mathcal{Q}_C \leftarrow \emptyset$; $\mathcal{Q}_D \leftarrow \emptyset$;
$(st, \mathcal{S}, \tau) \leftarrow \mathcal{A}^{\mathsf{OJoin}(),\mathsf{ODecaps}(),\mathsf{OCorrupt}()}(\mathsf{param})$;
$(H, K) \leftarrow \mathsf{Encaps}(\mathsf{EK}, \mathit{Reg}(\tau), \mathcal{S})$;
$K_b \leftarrow K$; $K_{1-b} \xleftarrow{\$} \mathcal{K}$;
$b' \leftarrow \mathcal{A}^{\mathsf{OJoin}(),\mathsf{ODecaps}(),\mathsf{OCorrupt}()}(st; \mathcal{S}, H, K_0, K_1)$;
`if` $\exists i \in \mathcal{S}, (i, \mathcal{S}, H) \in \mathcal{Q}_D$ `or` $i \in \mathcal{Q}_C$
`then return` 0;
`else return` b';

$\mathsf{OJoin}(i)$
$(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow \mathsf{Join}(\mathsf{msk}, i)$;
`return` upk_i;

$\mathsf{ODecaps}(i, \mathcal{S}, H)$
$\mathcal{Q}_D \leftarrow \mathcal{Q}_D \cup \{(i, \mathcal{S}, H)\}$;
$K \leftarrow \mathsf{Decaps}(\mathsf{usk}_i, \mathcal{S}, H)$;
`return` K;

$\mathsf{OCorrupt}(i)$
$\mathcal{Q}_C \leftarrow \mathcal{Q}_C \cup \{i\}$;
`return` usk_i;