The SKEMASNET project

Session Key Management in a Spontaneous Network

CS642 - Distributed Systems

PARK Hyunho - 20094045
RICCIARDI Gianni M. - 20096093
ALAUZET Pierre - 20096699

October 23rd, 2009

HGP Team

Background

The ***Skemasnet*** project goal is to find a secure protocol to share session keys during the merging of several spontaneous networks or the partitioning of an existing network into two or more sub-networks, independent from each other; this protocol aims to use pre-existing secure networks to share new session keys in an easy way, transparent for users.

Merging networks

In case of merging networks, one user is chosen in each network in order to manage the creation of the new key; all delegated users meet face to face, create the new key and share it using a *secure* channel (e.g. infrared ports on their laptops or portable devices). Once the new key is available, each chosen user sends it to all users (multicast) belonging to his/her former network, using the latter.

In order to get the new key an attacker should have one of the keys used to establish one of the merging networks.

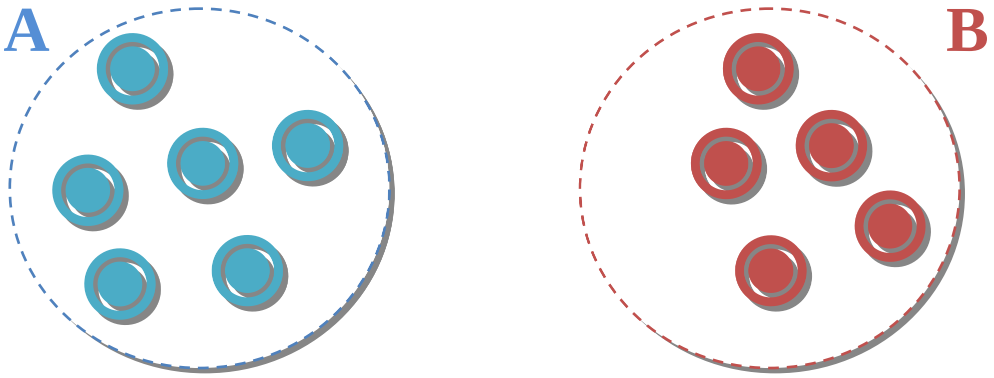
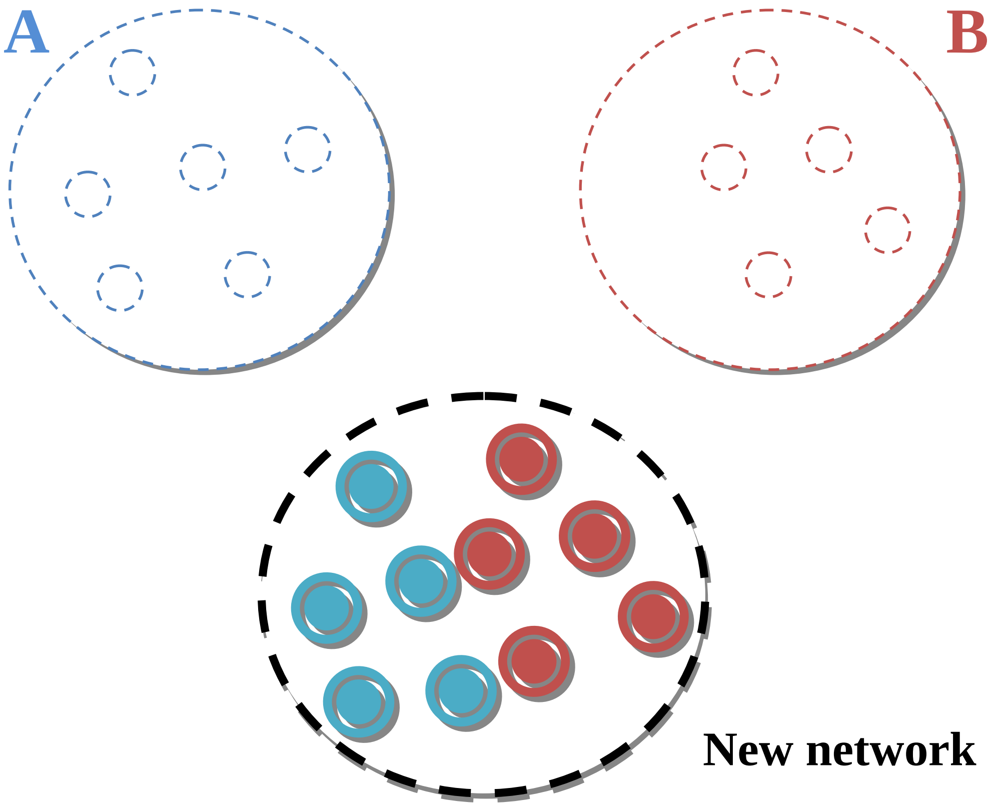
New key multicast will be available till the former networks are not running anymore. Transition period will be put in place during which both keys (previous networks keys and new single network key) are valid. Users will benefit of a sharing time in order to join the new network. This time could be about 5 or 10 minutes for example.

After the time limit, users who did not join the new network will be isolated. But each host owning the new key is able to share it again using the aforementioned *secure* channel: in the case of single isolated users they need to ask for the new key to a connected user (i.e. a user able to use the new network); instead in case of a group of isolated users, still using the previous network, they can use the same procedure described above in a recursive way, delegating someone to get the new key and then getting it from him/her using the still-running network.

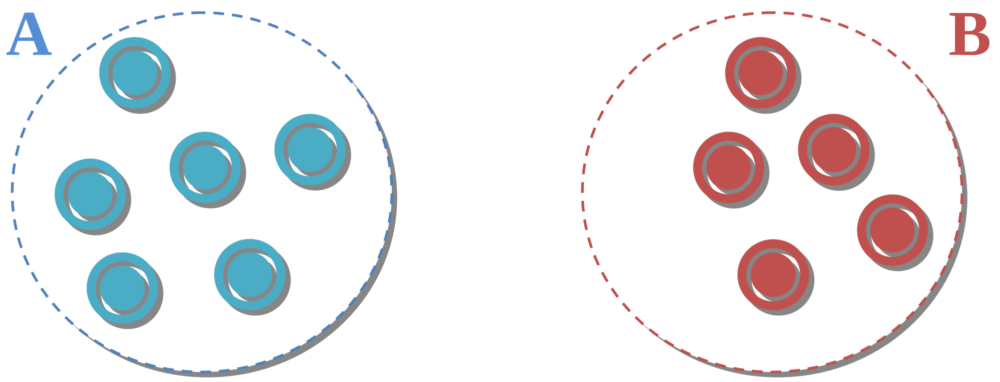
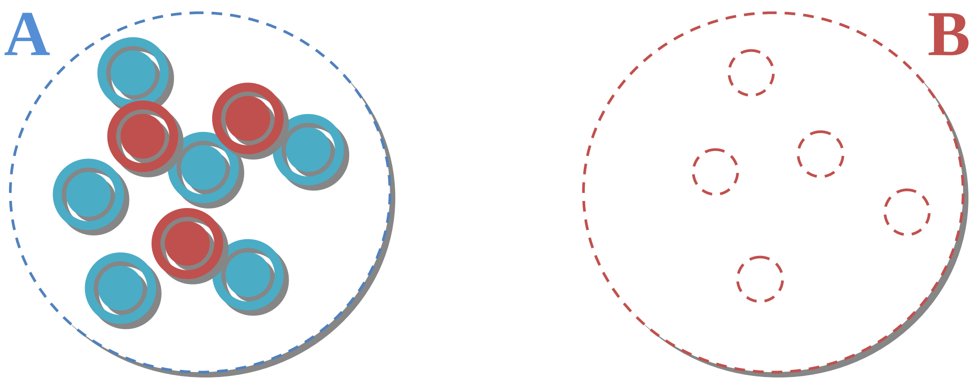
We are considering 2 cases of merging networks:

1. **Merging of networks A & B and creation of a new network**
2. **Group B joins the group A**

Case 1: *Merging of networks A & B and creation of a new network*

Initial situation	
After merging	

Case 2: *Group B joins the group A*

Initial situation	
After merging	

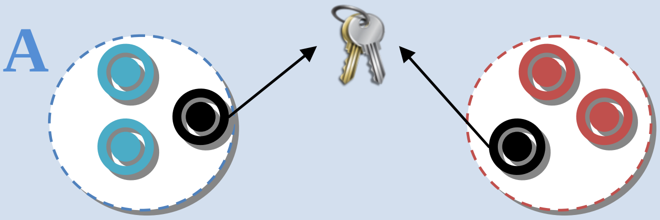
Let us give the scenario we want to follow for the merging of two networks:

Preface

Decision making came from human interactions.
Two groups meet and decide to merge their networks and then they choose two delegates (one per network).

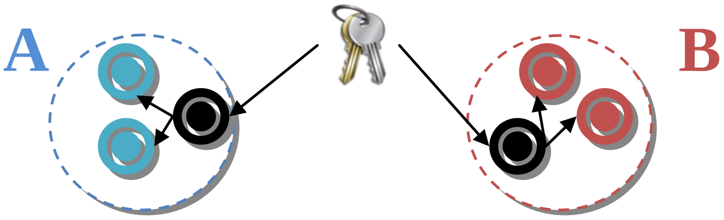
Initial Phase

Delegates create a new session key (or choose a key among two used keys)



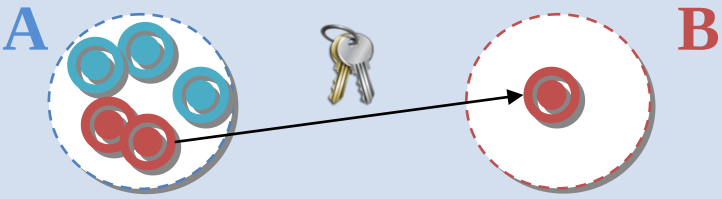
Propagate Phase

Each delegate propagates the new key to the members of his/her pre-existing network through the network itself.



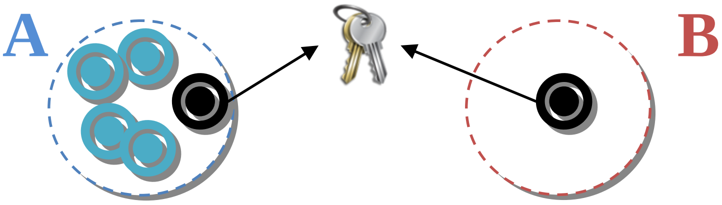
Suspend Phase

For a certain period (up to users or up to applications), members of the new network maintain pre-used session keys, together with the new one; if a node, member of a pre-existing network, that could not receive the new session key for some reason (ex. Physically isolated), tries to communicate with a node of its pre-existing network, it can receive the new key from the node.



Waiting Phase

If a new node requests to join the new network, it is performed as a common joining process.



Separating/partitioning networks

A pre-existing network can be partitioned in several ways. We thought about three different possibilities:

1. **Drop a secure network and create new independent networks.** This solution is not convenient for users and does not consider shared resources that will be lost.

2. **Use the existing network in order to create, propagate and share session keys for new networks, before leaving the former network and use only the new ones.** It is actually the fastest and most convenient solution for users, even if we are aware of a security issue: during the propagation of a new session keys using the pre-existing network, users not belonging to the group creating the new sub-network are able to sniff the traffic and to potentially get the new key.

3. **Use the existing network as well as a group key creation mechanism to obtain a new key before leaving the former network and use only the new one.** In this case, thanks to group key creation, only users belonging to a separating group can share the new key, thus fixing the security problem described in the previous case.

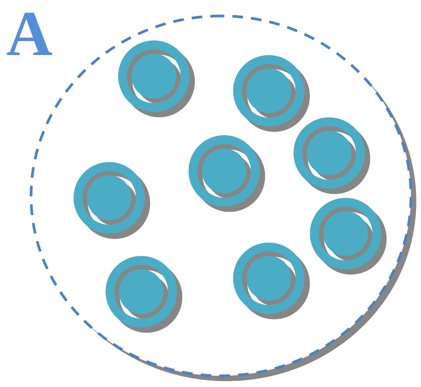
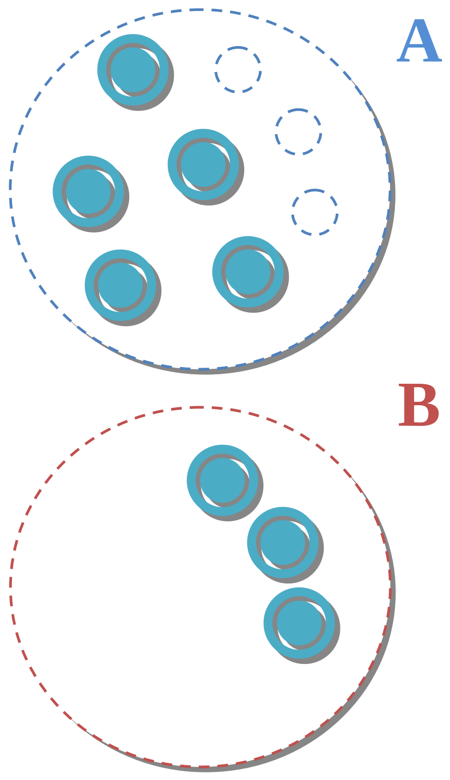
Among these solutions, we chose the **second one**, but still **considering the third idea as a possible way to improve security**. The scenario we are considering is the following:

In an already deployed spontaneous network, a subset of users decides to create their own network, independent from the existing one.

One user is delegated to fill a list of all users belonging to the new network; then a new session key, used to enable the new network, is created and propagated only to users belonging to the group, making them able to communicate independently of the former network. He will use the channel that everybody knows in order to propagate the new key.

Other users are still able to use the pre-existing network. The previous network will still exist if some others users are still using it. The new network will be totally independent and will not stop the previous network. Of course, once no one is using the previous network, it will automatically be deleted. The same process will occur with the new network.

Since the previous network, available to all users, is used to share the new key, a malicious user not belonging to the separating group could eavesdrop and get that key fraudulently. Anyway we suppose a **certain level of trust among *all* users belonging to the original network**, and that no classified or invaluable data are shared onto this kind of network.

Initial situation	After separation
	

Let us give the scenario we want to follow for the separation of a network:

Preface	*Decision making came from human interactions.* *A user decide to quit the network with some other users and decide to create a new one*
Initial Phase	Get a member list for new partitioned network. Create a secure network using GKA protocol between the members.
Propagate Phase	Same as merging
Suspend Phase	Same as merging
Waiting Phase	Same as merging