Assignment 2

### *Part II*

## Problem 1

Problem summary:

* Call problem *SUS-COAL*
* User, *u*
* Minute, *m*
* IP address, *I*(*u*, *m*) or ∅ if no entry.
* Server attacked and attacker accessed *i* distinct IP addresses over *t* consecutive minutes. Therefore, in minute 1: attacker accessed address *i*1, in minute 2: accessed address *i*2, and so on until minute *k*.
* Need to analyse. No single user *u* so that *I*(*u*, *m*) = *im* for each minute m from 1 to *k*.
* Subset of users *S* are suspicious if, for each minute *m* from 1 to *t*, there is at least one user *u* in *S* for which *I*(*u*, *m*) = *im*.
* Given a collection of all values, and a number *l*, is there a suspicious coalition of size at most *l*?
* Prove NP-complete

To prove a problem is NP-complete, we have to prove that it is in NP and that every problem in NP can be reduced to the provided problem. We want to be able to prove that this problem is so hard that if we could solve it, we could also solve every other NP-complete problem.

We first show that this problem belongs to NP. Given some instance of *S*, *I*, *l*, and *u*, we can provide a certificate. To do so, we iterate over each minute, *m*, and if there is a user, *u*, such that *I*(*u*, *m*) = *im*, then we return *true*. However, if none is found we return *false*. This process can be done in polynomial time O(*l* \* *t*).

We will prove this problem is NP-complete by reducing it to VERTEX-COVER problem, a known NP-complete algorithm. Therefore, we want to prove VERTEX-COVER ≤P SUS-COAL.

We are given VERTEX-COVER instance which is a graph, *G* = (*V*, *E*) and an integer *j* representing the size of vertex cover. Given our graph, *G* = (*V*, *E*), we can

* Set *S* = *V*
* | *E* | = *t* IP addresses
* We then map each (*u*, *u`*) ⊆ *E* to a minute *m*
* *I*(*u*, *m*) = *im*

We now have *S*, *I*, and *l* for SUS-COAL.

We can now claim that our set *S* has a size of at most *l* if, and only if, G contains a vertex cover size of at most *l*. To prove this, we’ll demonstrate both directions of the statement.

## Problem 2

For both problems, we want to check its membership in NP. We know the set of conversations, *S,* is verifiable in polynomial time since we just check that |*S*| ≥ *k*, *S* ⊂ *E*, and that no two edges in *S* are neighbours.

***3-SAT***

When provided a 3-SAT formula of *n* variables and *m* clauses, we can convert this to a graph *G* using a set of gadgets. Each 3-SAT variable gadget sub-problem has a central vertex that can talk to one but not the other of its set of neighbours.

Something

***Vertex Cover***

Provided an instance (*G* = (*V*, *E*), *k*) of VERTEX-COVER. We can convert it to an instance of CELLPHONE-CAP problem, (*G*` = (*V*`, *E*`), *k*`).

For each vertex in the set of vertices, *V*, we define a new vertex, *v*`. Therefore, we have a new graph such that every vertex in *G* exists in *G`*. We then define each vertex, *v*, a partner vertex, *v`*.

V` = V ∪ { v` | v ⊆ V } and

E` = E ∪ { (v, v`) | v ⊆ V }

## Problem 3

***Part A***

Something

***Part B***

Something

***Part C***

Something

***Part D***

Something