Anna Zych

New keys to data security

How to give the right person access to the right part of a database? Anna Zych looks at the structure of the organisation and assigns private encryption keys to all users. Based on their position within the organization, a larger or smaller part of the information opens up for them. Encryption and a smart way of generating the keys keep disabuse out.

‘As long as no hard disk gets into the wrong hands or a server isn’t hacked, the current approach to access control is pretty safe. Within an organisation, you can determine who gets access to which part of the information in a database, using access policies. The access rights of the administrative employee will differ from the rights of the department manager, the general manager gets access without too many limitations. Based on the hierarchy within the organisation, access policies like that can be defined. Looking at larger organisations, this can be really complicated. You have to verify, for example, that there are no conflicting policies. Using intelligent filters and algorithms, it is already possible to protect the data. I want to go a step further, however. If a hacker now manages to get access to the server or the hard disk, not only the data can be found but the access list as well. The information may get out on the street then, no matter how carefully you’ve built your access policy. Especially when privacy is involved, this is disastrous. Take for example an electronic patient file that should be visible for a very limited number of people.’

‘That’s why I use encrypted data to start with. Every person within the organization gets a private key to decrypt his or her part of the database. The server contains no list or access code whatsoever. Without a key, the data makes no sense for a hacker. Assigning the right keys to the right persons, giving one person more rights than the other, that’s the following step. How can someone, higher in the hierarchy, reach the information available to the lower levels? By giving him a list of keys, you could say. But if there’s one thing we want to avoid, it’s that a list of keys can become public in any way. That’s why I am developing a way to obtain a key that’s one level below your level, by using your own private key. It’s not just your own key you need, you have to use a second key of the same hierarchical level as yours. Every ‘child key’ has to have two parents, in this way. Working this out further, you get a V-shaped hierarchy. At the lowest level, the employees only have their private keys.’

‘Of course, this V-hierarchy doesn’t often correspond to the hierarchy you see in a real life organization. There will be cross connections between levels, making things really complicated. That’s why I have to find out the hierarchy in the real organization, and translate this arbitrary from to a V-shape using graph transformation. One of the things you will notice then, is that not every employee, every node in the network, has two parent nodes. In those cases we define an extra virtual node. We add a key to be able to find the lower key. The number of public keys has to be as small as possible, however: information that gets public makes the system more vulnerable. Apart from that, you want to adapt to organizational changes quickly. Reassign the keys whenever necessary, without losing data security. It has to remain waterproof!’

‘That’s quite a puzzle to solve. I can fully put my interest in order theory into this, as well as cryptography and algorithmics. I particularly enjoy the graph transformation part of this problem. After developing software for transforming the hierarchy and finding a method for key generation, we want to apply this method to real access control. It seems to be a very promising approach.’