Computers and Electrical Engineering 47 (2015) 318–326

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Password systems: Design and implementation ☆

Gianluca Dini, Lanfranco Lopriore *

Dipartimento di Ingegneria dell'Informazione, Università di Pisa, via G. Caruso 16, 56126 Pisa, Italy

ARTICLE INFO

Article history:
Received 28 April 2014
Received in revised form 6 February 2015
Accepted 6 February 2015
Available online 5 March 2015

Keywords:
Access right
Distribution
Key
Password
Revocation

ABSTRACT

Critical infrastructures require protection systems that are both flexible and efficient. Flexibility is essential to capture the multi-organizational and state-based nature of these systems, efficiency is necessary to cope with limitations of hardware resources. To meet these requirements, we consider a classical protection environment featuring subjects that attempt to access the protected objects. We approach the problem of specifying the access privileges held by each subject. Our protection model associates a password system with each object; the password system features a password for each access privilege defined for this object. A subject can access the object if it holds a key matching one of the passwords in the password system, and the access privilege corresponding to this password permits to accomplish the access. Password systems are implemented as hierarchical bidimensional one-way chains. Trade-offs are possible between the memory requirements for storage of a password system and the processing time necessary to validate a key.

1. Introduction

Critical infrastructures are essentially physical processes controlled by networked computers. They are usually as vulnerable as any other interconnected computer system, but their failure may have a high socio-economic impact [1]; furthermore, they present distinguishing features that make their protection a problematic challenge [2]. In critical infrastructures, the use of wireless sensor and actuator networks (WSANs) is becoming pervasive, and consequently the network boundaries of the system become blurred. The integration of multi-hop WSANs with supervisory control and data acquisition (SCADA) systems to monitor critical infrastructures is considered a promising approach [3], which extends the extent of SCADA systems and represents a cost-effective solution to the problem of limited deployment flexibility.

However, the integration of WSAN with SCADA poses new cybersecurity challenges, e.g. an adversary equipped with a radio transceiver can access the wireless medium and attack sensors and actuators at little effort, to eavesdrop their state, alter their set-up, and issue fraudulent commands. Sensors and actuators are usually resource-scarce devices, and this precludes utilization of off-the-shelf protection solutions, e.g. digital signatures and trusted hardware. Furthermore, a protection system for critical infrastructures is required to comply with a multi-organizational nature and different operational states. In fact, subjects from distinct organizations are often involved in a single critical infrastructure. Each subject executes operations in the infrastructure, and almost all operations are based on a state model. It follows that the actions a subject can undertake in a given state may be forbidden in a different state.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Joaquin Garcia-Alfaro.

* Corresponding author.
E-mail addresses: g.dini@iet.unipi.it (G. Dini), l.lopriore@iet.unipi.it (L. Lopriore).

http://dx.doi.org/10.1016/j.compeleceng.2015.02.012
0045-7906/

In the design and implementation of a protection system, these complex aspects of a critical infrastructure should be taken into careful account. In particular, the multi-organizational and state-based characteristics imply that the protection system should be able to grant and revoke access privileges with flexibility and efficiency. Fine-grained forms of protection should be supported at level of a single object (a data item, as well as a device). The requirements of the protection system in terms of computing power and storage overhead should be kept to a minimum to comply with the limited resources available in the physical objects. Access control has been listed among the technical and management activities aimed at limiting or containing cybersecurity events that are common across critical infrastructure sectors [4]. In particular, accesses to assets and associated facilities must be limited to authorized users, processes, or devices, and to authorized activities and transactions. A suitable management of access permissions is required to incorporate the principles of least privilege and separation of duties.

In the following, we shall refer to a classic protection model featuring active entities, the *subjects*, which perform access attempts to passive entities, called *objects* [5,6]. Objects are typed; the type of a given object states the values that can be assumed by this object and the operations that can be applied to these values. The type also states the set of *access rights* and the associations between the operations and the access rights. In order to access an object G of a given type T to perform one of the operations of this type, a subject must possess the access rights permitting successful execution of this operation.

A basic problem in every protection model is to specify the access rights that each subject holds on the protected objects. In a classical approach, a set of *passwords* is associated with each object, and each password corresponds to an access privilege defined in terms of one or more access rights [7–9]. Subject S is entitled to access object G if it possesses a key k matching one of the passwords associated with G. The key certifies that the subject holds the access rights corresponding to the matching password, and allows the subject to accomplish the operations made possible by these access rights. Keys are protected from forging by the password length [10]. If passwords are large and chosen at random, the probability to guess a valid key is vanishingly low.

1.1. Key distribution and derivation

Subject S that possesses key k for object G can transfer the key to another subject S'. Consequently, S' acquires the access rights for G that are granted by k. A salient problem of key distribution consists in allowing subject S to grant S' only a part of the access rights included in the original key k. *Key derivation* is the process of transforming key k into another key k' that grants a weaker privilege. This means that the new key k' matches a password corresponding to the weaker privilege.

In a centralized approach, key derivation can be obtained by associating a password manager with each object. The password manager receives a key corresponding to a given password and returns a key corresponding to a password with reduced access rights. This approach implies interaction between a subject and the password manager. In a different approach, the protection system includes a key derivation mechanism that allows subjects to transform keys into weaker keys autonomously.

1.2. Key revocation

Ease of access privilege distribution certainly was one of the main reasons for the introduction of password-based protection. A subject that receives a key matching a given password is free to propagate the key further to other subjects; a result of this type will be simply obtained by a key copy. It follows that keys tend to propagate throughout the system.

A related problem is that of *key revocation* [11–14]. A key is revoked when it can no longer be used for successful object access. The protection system should support a key revocation mechanism, and a *revoke* access right, so that a subject that holds this access right for given keys can revoke these keys. Revocation should be selective, so that only a subset of the keys distributed for a given object are revoked. Revocation should transitively extend throughout the system to all the copies of the revoked keys, as well as to all the keys derived from the revoked keys, and their copies.

Of course, in a password-oriented system, a simple solution to the key revocation problem is password replacement. If we change the password for a given access privilege, all the keys matching this password are revoked. This solution does not meet the requirement to extend revocation automatically to all the keys derived from the key being revoked. It follows that a subject can circumvent revocation by taking advantage of the derived keys.

In this paper, we present a model of a protection system based on typed objects, passwords and keys. Our model was designed to meet the following requirements:

- A subject that holds a given key should be given the ability to derive new keys with reduced access privileges. The entire key derivation process should be local to the subject, with no need for intervention of a centralized password manager.
- Forms of selective key revocation should be supported. Revocation should be transitive with respect to key derivation, so that revocation of a given key k implies revocation of all copies of k, as well as of all the keys derived from k, and their copies.
- At the implementation level, trade-offs should be possible between the memory space necessary for password storage and the processing time for key validation, so that if low memory cost is not a stringent requirement, the time necessary to validate a key can be kept to a minimum, for instance.