

Authentication

A model of human – machine authentication

F. Chabaud and O. Grumelard¹

florent.chabaud
olivier.grumelard } @sgdn.pm.gouv.fr
astec

SGDN/DCSSI/SDS/AsTeC
51 Bd La Tour Maubourg
75700 Paris SP
France

Abstract

We introduce a model of authentication between a person and an information system. This model makes a clear distinction between cryptographic operations, which occur between automated entities that are part of the information system, and what we define as unlocking operations, which are more specifically performed by human beings.

Résumé

Nous présentons un modèle de l'authentification entre une personne et un système d'information. Cette modélisation permet de distinguer les opérations proprement cryptographiques, qui s'effectuent entre certaines des machines constituant le système d'information, et des opérations plus particulièrement réalisées par la personne, que nous définissons comme des opérations de déverrouillage.

¹ **Acknowledgments:** This work has been greatly improved through collegial work at DCSSI labs. We wish to thank all the people who took part in the elaboration of the DCSSI authentication reference document, from which the concepts presented here were extracted. Special thanks go to Pascal Chour, Rémy Daudigny, Loïc Dufлот, Fabien Germain, Eliane Jaulmes, Véronique Joubert, Philippe Le Moigne, Gwenaëlle Martinet, Michel Mitton, Louis Mussat, Guillaume Poupard and Philippe Wolf.

1 Introduction

Authentication is a widespread function in information systems, and it is often of critical importance for security purposes. Its main objective is to give a certain guarantee as to the truthfulness of an identity claimed by an entity, be it a person or a machine. It is important to note that, in this document, a claimed identity is supposed to be already available. We therefore only have to check that this claim is sincere.

Authentication may rely on different kinds of mechanisms that neither offer the same guarantees, nor present the same robustness. The purpose of this document is to introduce a model that may be used to describe or evaluate those mechanisms.

Authentication itself may be used for several higher-level purposes. It may be used to help enforce access control to information, buildings, or more generally goods of an information system. Imputation mechanisms may rely on it too, e.g. for logging or charging purposes, as it provides a stronger guarantee regarding claimed identities. Authentication may also be used to fulfil a combination of both functions.

Replacing the authentication function within its context of use raises the issue of proper integration within its environment in order to fulfil its true goal, that is, the attribution – for authorization or imputation purposes – of an action to its actual perpetrator or, in other words, that the entity behind the action is the one that was authenticated.

There are two ways of handling this integration:

- signed acts, for which the link between authentication and an action is immediate;
- authenticated sessions, for which authentication is performed once at the beginning of a session, before the first action is allowed to take place, and which therefore requires a means of tracking further actions back to the opening phase.

It should be noted that this document does not target the issues of electronic signature or registration of a user within an authentication system.

It is traditional to build authentication schemes over one or several factors from the following list:

- what one knows – for example, a password;
- what one has – for example, a smartcard;
- who one is – measured, for example, from a fingerprint;
- what one can perform -- for example, a handwritten signature.

It should be noted that these last two authentication factors are clearly anthropomorphic and cannot be used to authenticate automated devices. We will thus make separate models for machine-to-machine authentication and human-machine authentication.

We will also assume that the authentication process can only take place if some kind of information is already shared between the two parties.

2 Authentication function model

2.1 Authentication process model

The implementation of access control and imputation functions involves:

- a supplicant, that wishes to carry out actions but has to prove its identity for that purpose,
- a monitor, that may allow the actions to take place provided it has verified the identity of their perpetrator.

The sequence of actions is sent through a communication channel between the supplicant and the monitor. The authentication should allow the monitor to reliably track messages received on this canal back to the identity of the supplicant. For this purpose, the channel must be activated in a way that is directly related to the authentication phase.

The time during which the canal is used by the supplicant constitutes an authenticated session. This session can be terminated:

- at the request of the supplicant, through explicit disconnection,
- by decision of the monitor, if it considers itself no longer able to guarantee the relationship between the channel and the identity of the supplicant.

In many cases, the authentication phase will secure the channel through a cryptographic key exchange. The session itself will use these keys to preserve the integrity and, if need be, the confidentiality of the communication flow.

We therefore identify:

- an initial state, unauthenticated, where the monitor forbids all actions;
- a connection phase, which opens the channel and authenticates the supplicant;
- an authenticated state of variable duration, namely the authenticated session, during which actions are authorized by the monitor;
- a disconnection phase, which brings the system back to its initial state.

All these states may generate errors and corresponding alarms. Transitions between states depend on the context. Successive states of an authenticated session are shown on figure 1.

2.2 Machine-to-machine authentication

From now on we will consider three information systems (IS):

- the local IS,
- the access IS,
- the trusted authentication IS.

The local IS is the supplicant, i.e. the machine that gets authenticated by the access IS, which is the monitor.

The model described on figure 2 also introduces a possible trusted authentication IS – which may or may not be used, – with which both information systems interact. This could be, for example, an authentication server that would authenticate the local IS on behalf of the access IS and whose verdict would in turn be authenticated by the access IS.

On this figure, arrows represent potential authenticated channels between entities. The main authentication channel (red arrow) connects the supplicant to the monitor. In most cases, the other channels must be authenticated in order to secure the authentication on the main channel. This may be achieved through communications, but may also be the result of trust relationships established, for instance, through prior enrollment. Such interactions are not mandatory.

Example 1 - IPSEC with manual keying

As an example, this model may be applied to a client and a file server linked by an IPSEC virtual private network (VPN) that is configured manually with a shared secret key. The server instantiates the access IS that checks access permissions to the files, while the client is the local IS. As all configuration operations are done manually, there is no trusted authentication IS.

Example 2 - IPSEC with public key infrastructure

This model may also be applied to a similar situation involving a client and a file server linked by an IPSEC VPN, established using a public-key infrastructure (PKI) and a signed Diffie-Hellman key exchange protocol. The PKI is an additional entity which instantiates the trusted authentication IS. It takes part in the mutual authentication scheme through prior certification of both the client and the server.

Example 3 - Physical access control

In order to check that the model is relevant for a whole range of situations, we may consider a totally different example: physical access control using an RFID tag. This includes:

- a local IS, the RFID tag, which is the supplicant,
- an access IS, the locking device on the door, which is the monitor,
- a trusted authentication IS, the server that manages access rights according to the identity announced by the tag.

This example clearly shows that there is no authentication of the RFID tag holder. Only the tag is authenticated. Moreover, several scenarios could be proposed for access rights verification:

1. The tag authenticates itself to the trusted authentication IS, that checks access rights and signals the access IS to open the door. This signal must be "authenticated" either by a cryptographic mechanism or by the physical security of the connection between the trusted authentication IS and the access IS.
2. The tag authenticates itself to the access IS, which then asks the trusted authentication IS whether the announced identity is authorized or not. Again, the security of the communication between the access IS and the trusted authentication IS could be enforced by various mechanisms.

2.3 Human-machine authentication

It is difficult to implement the authentication of a person by an information system in a direct way. Indeed, machines should consider only cryptographic mechanisms to be reliable, while human beings cannot directly use them.

In fact, all direct "authentication" mechanisms usable by human beings are characterized by their subjection to replay attacks. Hardly ever does a person change passwords after each use and biometrics- or behaviour- based mechanisms actually need replay ability in order to work. To our knowledge, there is no human-usable "authentication" mechanism that would be robust to replay attacks.

To avoid any confusion, we will call these techniques "unlocking" mechanisms. Indeed, these mechanisms often give access to resources that are themselves subject to an access control.

Examples of unlocking operations, characterized by replay ability, are:

- entering a password, which unlocks a computer,
- showing off a personal badge, which "unlocks" it by making it accessible to verification operations,
- inserting a removable device, which gives access to its data,
- typing in a PIN code, which activates the functionalities of a smartcard,
- displaying a biometric characteristic,
- etc.

The model described here complements the previous one by adding the user (see figure 3). In this model, the user is the one who gets authenticated, but the access rights will be granted to the local IS to carry out actions on behalf of the user. Authentication therefore takes place between the local IS and the access IS, but it requires prior unlocking of the local IS by the user.

Example 4 - File server – password-based unlocking

To start with, we can consider the file server example again, and add the user of the local machine. This user will locally unlock his or her machine with a password, which will allow her to use secret data stored on the local machine, whose access is controlled by the local operating system. The local machine thus plays the role of a local IS. It also stores secret data belonging to the user that constitutes the user's authentication token. The local IS will use this data to authenticate the user with the access IS (the server), which will grant her access to the files.

Example 5 - Physical access control

We can also consider the physical access control example, this time focusing on the user. In fact, the user is the supplicant who wants to open the authenticated canal: the doorway. For that purpose, she uses a token, the tag, which will request the opening of the door on her behalf. The monitor, the access IS, is by definition in charge of guaranteeing the authenticity of the actions sent through the canal. This is the reason why it may, for instance, have to close the door after a few seconds if it considers itself no longer able to certify the correspondence between the identity that was authenticated and the person that is actually able to cross the door.

In this example, several unlocking mechanisms may be considered:

- showing off the RFID tag is an unlocking mechanism, since it allows the tag to be activated;
- if the support implements a PIN code, typing in the PIN code is an unlocking mechanism;
- biometric characteristics can similarly be used to unlock the tag;
- one could also consider a system based on one time passwords generated by the tag and, in this case, the unlocking action could either be the activation of the generation mechanism and/or the typing of the generated password.

This example illustrates the fact that all unlocking mechanisms are not equally robust, especially against the threat of lost or stolen tags.

Example 6 - File server – authentication support-based unlocking

As a third example, we may consider again the case of a local client accessing a file server. This time we introduce a USB mass storage device which is used to store secret data. In this case, authentication data can only be accessed by the local IS if the support is present, which constitutes the unlocking of the local IS. Furthermore, if the support is a smartcard, then the unlocking can be more robust by letting the resource carry out some of the cryptographic operations. In such a case, the local IS does not even have access to the authentication keys used to open the canal. This unlocking mechanism can

be further reinforced if the smartcard itself has to be unlocked by the user, especially if this unlocking mechanism is independent from the local IS.

2.4 Authentication between remote users

Authentication between two remote users is achieved straightforwardly through symmetrisation of the previous model (see figure 4). In order to mutually authenticate, both users will rely on the features of their respective IS, that will perform mutual authentication on their behalf (green arrow). The need to unlock each IS will prove the other IS that the authentication is valid (pink arrows). Both users being confident in their own IS, this authentication scheme mutually authenticates the users (red arrows).

3 CONCLUSION

We introduced a model for authentication in information systems, which is based upon, on the one hand, an authenticated canal and authenticated sessions and, on the other hand, the so-called unlocking mechanism. The main feature of this model is that unlocking operations, which are intrinsically replay-vulnerable, are only performed locally, while remote operations rely exclusively on cryptographic protocols suitable for authentication between information systems.

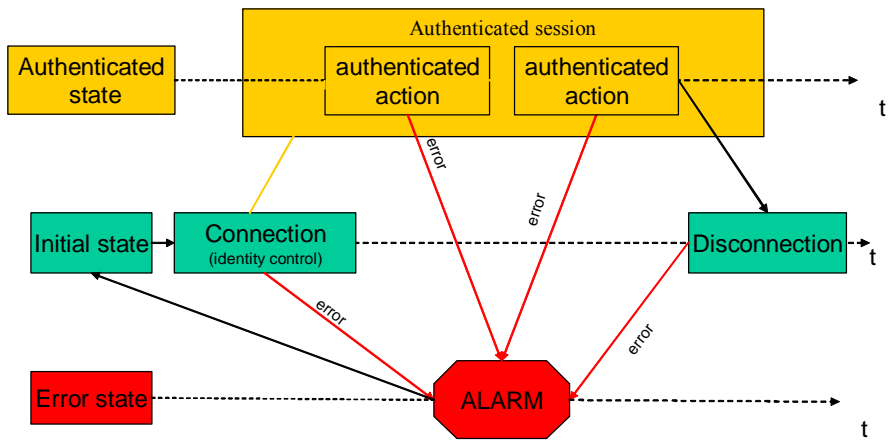


figure 1 - States of an authentication process

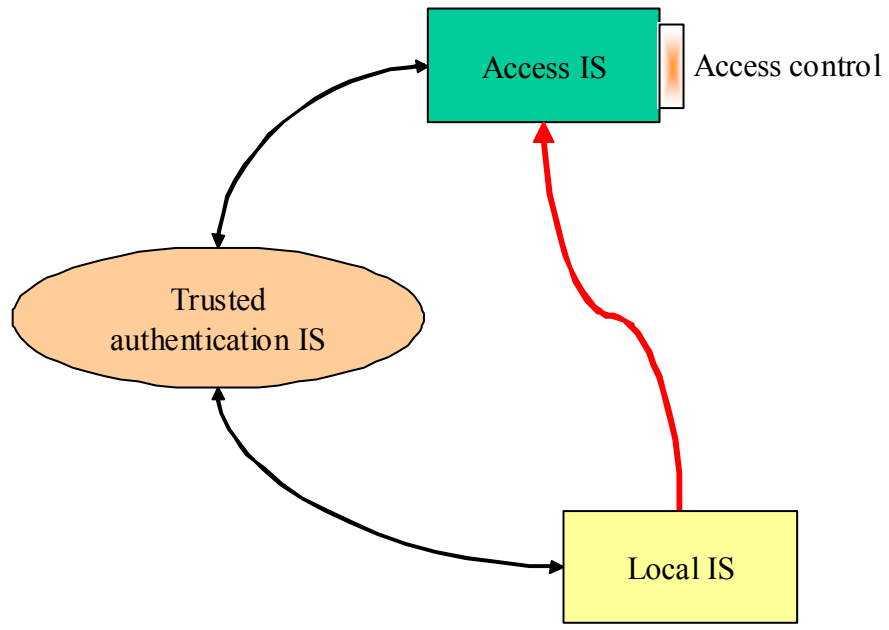


figure 2 – Machine-to-machine authentication

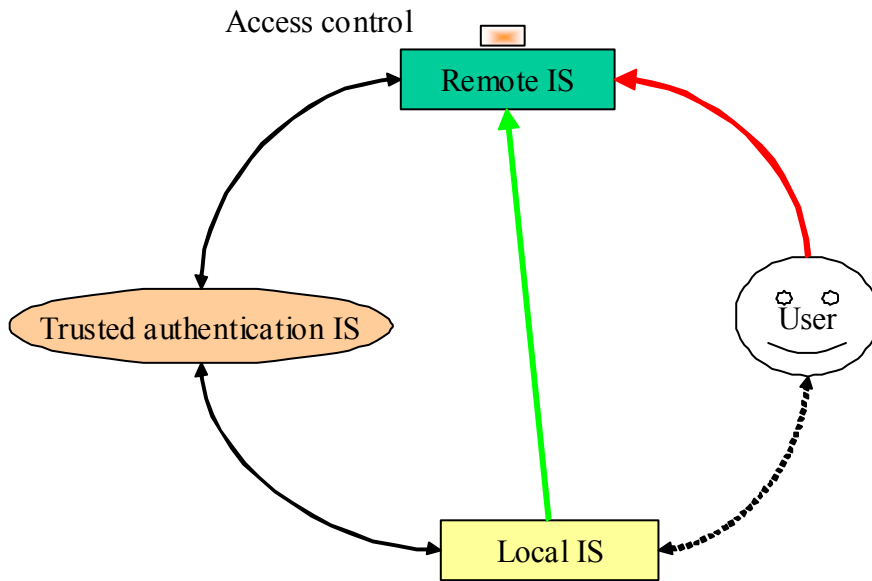


figure 3 – Remote human - machine authentication

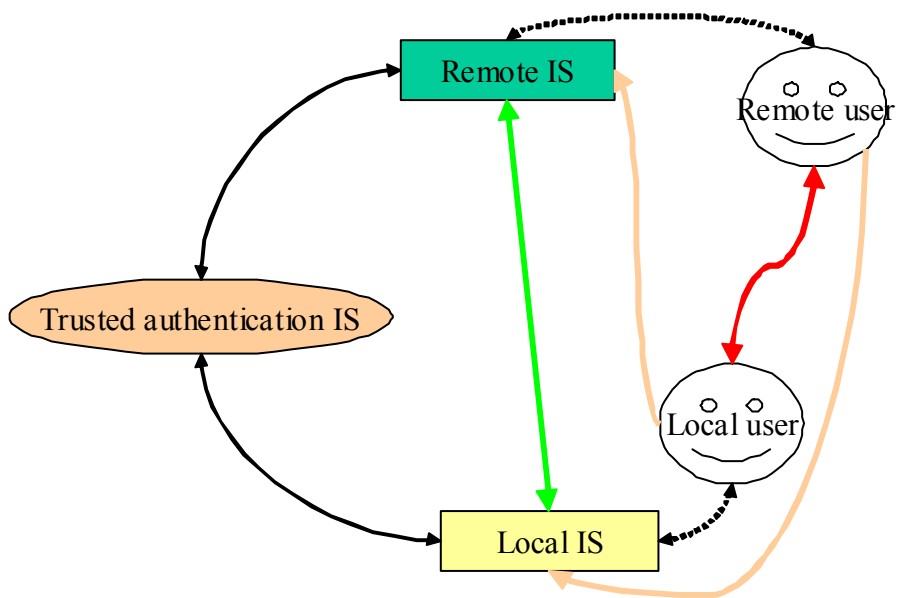


figure 4 - Remote user-to-user authentication

1 Introduction

L'authentification est une fonction de sécurité largement utilisée dans les systèmes d'information et qui revêt souvent une importance cruciale. Elle a pour objectif d'obtenir une certaine garantie quant à l'identité annoncée d'une entité, personne ou machine. Il est important, dès à présent, de noter que dans ce document, l'identification est considérée comme acquise. La fonction visée est celle qui consiste à vérifier la réalité de cette identité.

L'authentification peut reposer sur des procédés différents qui ne présentent pas tous les mêmes garanties, ni la même robustesse. L'objectif de ce document est de présenter une modélisation permettant de décrire ou d'évaluer les mécanismes.

L'authentification en tant que telle a peu de sens. Elle vise soit à contrôler l'accès à des informations, des locaux, plus généralement des biens d'un système d'information, soit à garantir une imputabilité (pour la journalisation, la facturation, etc.) avec vérification forte de l'identité affichée, soit à assurer une combinaison de ces fonctions.

Le remplacement de la fonction « authentification » dans son contexte permet en particulier d'introduire la question de l'intégration de cette fonction dans son environnement dans le cadre de l'objectif recherché, à savoir l'attribution (pour autorisation ou imputation) d'une action à son auteur réel ou, dit autrement, que l'entité qui agit est bien celle que l'on a authentifiée.

On distingue deux grands types de solutions :

- l'acte signé pour lequel le lien entre l'authentification et l'action est direct ;
- la session authentifiée, pour laquelle l'authentification intervient ponctuellement en début de session, avant la première action, et qui nécessite par là même une traçabilité entre l'ouverture et le déroulement de la session.

Notons tout de suite que la problématique liée à la signature électronique n'est pas l'objet de ce document, pas plus que l'enregistrement éventuel de l'utilisateur dans le système d'authentification.

Il est traditionnel de faire reposer la fonction d'authentification sur un ou plusieurs éléments parmi :

- ce que l'on sait (par exemple, un mot de passe) ;
- ce que l'on a (par exemple, une carte à puce) ;
- ce que l'on est (par exemple, une empreinte digitale) ;
- ce que l'on sait faire (par exemple, une signature manuscrite).

Notons que ces deux derniers éléments d'authentification sont clairement anthropomorphes et ne s'appliquent pas à des dispositifs automatiques. Nous distinguerons donc deux modèles selon que l'authentification aura lieu entre machines ou s'il s'agit de l'authentification d'une personne vis-à-vis d'une machine.

Nous supposerons également que l'authentification ne peut s'effectuer qu'après partage préalable d'informations entre les acteurs concernés.

2 Modèle de la fonction d'authentification

2.1 Modèle du processus d'authentification

La réalisation des fonctions contrôle d'accès et imputation fait intervenir :

- un demandeur, qui souhaite effectuer des actions et doit pour cela prouver son identité,
- un receveur, qui peut permettre les actions, en devant au préalable vérifier l'identité de leur auteur.

La suite des actions circule sur un canal reliant le demandeur au receveur. L'authentification permet de relier de façon fiable, pour le receveur, les messages circulant sur ce canal à l'identité du demandeur. L'activation du canal doit par conséquent s'effectuer en relation directe avec l'authentification.

Le temps d'exploitation du canal par le demandeur constitue une session authentifiée. Cette session peut se terminer :

- à l'initiative du demandeur, s'il se déconnecte,
- à l'initiative du receveur, s'il estime qu'il n'est plus en mesure de garantir le lien entre le canal et l'identité du demandeur.

Dans beaucoup de cas, la phase d'authentification sécurisera le canal par un échange de clés cryptographiques. L'ensemble de la session utilisera ces clés pour se protéger en intégrité et si besoin en confidentialité.

On distingue donc :

- un état initial, non authentifié, dans lequel le receveur interdit les actions ;
- une phase de connexion, c'est-à-dire d'ouverture du canal, qui constitue l'authentification du demandeur ;
- un état authentifié d'une certaine durée, constituant la session authentifiée pendant laquelle les actions sont autorisées par le receveur ;
- une phase de déconnexion permettant le retour à l'état initial.

Tous les états peuvent potentiellement engendrer une erreur qui peut générer une alarme. Les transitions entre états, quant à elles, dépendent du contexte. Les états successifs de la session authentifiée sont présentés dans la figure 1.

2.2 Modèle d'authentification de machines

Nous allons distinguer dans la suite trois systèmes d'information (SI) :

- le SI local,
- le SI d'accès,
- le SI d'authentification de confiance.

Le SI local est le demandeur, à savoir la machine qui s'authentifie auprès du SI d'accès, le receveur.

Le modèle de la figure 2 introduit aussi un SI d'authentification de confiance éventuel (qui peut ou non être utilisé) avec lequel les deux systèmes d'information sont en interaction. Il s'agit, par exemple, d'un serveur d'authentification qui authentifierait le SI local pour le compte du SI d'accès et dont la réponse serait elle-même authentifiée par le SI d'accès.

Dans cette représentation, les différentes flèches représentent les canaux authentifiés possibles entre les entités. Le canal d'authentification principal (flèche rouge) relie le demandeur au receveur. Les autres canaux nécessitent dans la plupart des cas d'être authentifiés pour garantir la sécurité de l'authentification du canal principal. Ces interactions peuvent être des communications, mais aussi des relations de confiance établies, par exemple, par un enrôlement. Ces interactions ne sont pas obligatoires.

Exemple 1 – IPSEC avec mise à la clé manuelle

À titre d'exemple on peut chercher à appliquer ce modèle à un client et un serveur de fichiers reliés par une liaison IPSEC configurée manuellement à l'aide d'un secret partagé. Le serveur joue le rôle de SI d'accès contrôlant l'accès aux fichiers, tandis que le client est le SI local. La configuration étant manuelle, il n'y a pas de SI d'authentification de confiance.

Exemple 2 - IPSEC avec infrastructures de clés publiques

De même, on peut appliquer le modèle à une situation similaire impliquant un client et un serveur de fichiers reliés en IPSEC, mais cette fois-ci utilisant une infrastructure de clés publiques et un protocole d'échange de clés Diffie-Hellman signé. Le modèle comprend alors en plus l'infrastructure de clés publiques qui joue le rôle de SI d'authentification de confiance. Elle participe à l'authentification mutuelle par la certification du client et du serveur.

Exemple 3 - Contrôle d'accès physique

Pour vérifier le caractère général du modèle, on peut aussi envisager un exemple totalement différent de système de contrôle d'accès physique utilisant un badge sans contact. On y trouve :

- un SI local, le badge sans contact, demandeur,
- un SI d'accès, le dispositif de verrouillage de la porte, receveur,
- un SI d'authentification, le serveur qui gère les droits d'accès en fonction de l'identité annoncée par le badge.

On voit bien sur cet exemple qu'il n'y a pas authentification du porteur du badge. C'est uniquement ce dispositif qui est authentifié. En outre, on pourrait imaginer plusieurs scénarios de contrôle d'accès, par exemple :

1. Le badge s'authentifie auprès du SI d'authentification qui vérifie les droits d'accès et donne un signal au SI d'accès pour ouvrir la porte. Ce signal est dans ce cas « authentifié » soit par un mécanisme cryptographique soit par la sécurité physique de la connexion entre le SI d'authentification et le SI d'accès.
2. Le badge s'authentifie auprès du SI d'accès qui demande ensuite au SI d'authentification si l'identité annoncée est autorisée ou pas. Là encore, la sécurité de la transmission entre le SI d'accès et le SI d'authentification peut être assurée par divers mécanismes.

2.3 Modèle d'authentification homme – machine

L'authentification d'une personne vis-à-vis d'un système d'information est délicate à réaliser de façon directe. En effet, du point de vue de la machine, seul un procédé de nature cryptographique s'avère sûr, tandis que la personne, quant à elle, ne peut directement employer un tel mécanisme.

En effet, les procédés « d'authentification » directe d'une personne se caractérisent tous par la possibilité de rejeu. Il est rare qu'une personne change systématiquement de mot de passe à chaque utilisation et les procédés de nature biométrique ou comportementale utilisent tous, au contraire, le rejeu pour fonctionner. Il n'y a pas, à notre connaissance, de mécanisme humainement exploitable permettant une authentification sans rejeu.

Pour bien les distinguer, nous qualifierons ces procédés de déverrouillage. En effet, ces procédés permettent dans la plupart des cas d'accéder à des ressources soumises, là encore, à un contrôle d'accès.

Comme exemple de procédés de déverrouillage, caractérisés par la possibilité de rejeu, on peut citer :

- la saisie d'un mot de passe, qui déverrouille un ordinateur,
- la présentation d'un badge personnel, qui « déverrouille » ce dernier en le rendant accessible aux opérations de vérification,
- l'insertion d'un support amovible, qui donne l'accès aux données qu'il contient,
- la saisie d'un PIN code, qui active des fonctionnalités d'une carte à puce,
- la reconnaissance d'une caractéristique biométrique,
- etc.

Le modèle que nous décrivons complète le précédent en faisant apparaître l'utilisateur (voir figure 3). Dans ce modèle, c'est l'utilisateur qui s'authentifie, mais les droits d'accès qui seront ainsi ouverts le seront vis-à-vis du SI d'accès pour le SI local, lequel effectuera les actions au bénéfice de l'utilisateur. L'authentification s'effectue donc de machine à machine entre le SI local et le SI d'accès, mais grâce à un déverrouillage du SI local par l'utilisateur.

Exemple 4 - Serveur de fichiers – déverrouillage par mot de passe

Dans un premier exemple nous reprenons celui de l'accès d'un client local à un serveur de fichiers en introduisant l'utilisateur de la machine locale. Celui-ci va déverrouiller localement la machine avec un mot de passe, ce qui permet l'utilisation des informations secrètes stockées sur la machine locale dont le verrouillage est contrôlé par le système d'exploitation local. On voit que la machine locale joue le rôle de SI local. Elle héberge également les informations secrètes de l'utilisateur qui constituent le support d'authentification de l'utilisateur. Par la suite, le SI local utilisera ces données pour authentifier l'utilisateur auprès du SI d'accès (le serveur) et permettre d'accéder aux fichiers.

Exemple 5 - Contrôle d'accès physique

Comme deuxième exemple, reprenons l'exemple du contrôle d'accès physique, mais en faisant cette fois-ci apparaître l'utilisateur. C'est en effet lui qui est demandeur de l'ouverture d'un canal authentifié : la porte. Pour cela, il utilise un support, le badge, qui va obtenir pour lui l'ouverture du loquet de la porte. Le receveur, le SI d'accès, est par définition chargé de garantir l'authenticité des actions transitant par le canal « porte ». C'est la raison pour laquelle, il peut, par exemple, commander la fermeture du loquet au bout de quelques secondes s'il estime qu'il n'a plus de garantie de correspondance entre l'identité authentifiée au départ et la personne qui a effectivement la possibilité de traverser la porte.

Dans cet exemple, plusieurs mécanismes de déverrouillage sont possibles :

- la simple présentation du badge est un mécanisme de déverrouillage, puisqu'elle met le badge en situation d'activité ;
- si le badge dispose d'un code PIN, la saisie du code est une opération de déverrouillage ;
- de même, une caractéristique biométrique peut être utilisée pour déverrouiller le badge ;
- on pourrait aussi considérer un système de mots de passe à usage unique générés par le badge et, dans ce cas, c'est l'activation du mécanisme de génération et/ou la saisie du mot de passe généré qui constituent le déverrouillage.

On voit toutefois sur cet exemple que tous les mécanismes de déverrouillage n'ont pas la même robustesse, notamment par rapport à la menace de perte ou de vol du badge.

Exemple 6 - Serveur de fichiers – support d’authentification

Dans un troisième exemple, nous pouvons revenir sur l’accès d’un client local à un serveur de fichiers en imaginant un support de clés de type clé USB de stockage de masse, sans capacité de calcul. Dans ce cas, les données secrètes ne peuvent être accédées du SI local que si le support est présent, ce qui constitue le déverrouillage du SI local. Mais, si le support est une carte active, alors le déverrouillage peut être encore plus robuste en laissant la ressource effectuer une partie des calculs cryptographiques. Le SI local n’a alors jamais accès aux clés qui lui permettent d’obtenir l’ouverture du canal. On peut encore renforcer ce déverrouillage en demandant que la carte elle même soit déverrouillée par l’utilisateur, voire que le mécanisme de ce déverrouillage ne soit pas sous le contrôle du SI local.

2.4 Modèle d’authentification à distance

La modélisation de l’authentification à distance de deux personnes s’obtient naturellement en symétrisant le modèle précédent (voir figure 4). Pour que deux personnes puissent s’authentifier mutuellement, elles exploiteront les possibilités de leurs SI respectifs, qui réaliseront pour leur compte une authentification mutuelle (flèche verte). Le déverrouillage de ces SI constituera l’élément de preuve assurant l’authentification vis-à-vis des SI distants (flèches roses). Chaque utilisateur ayant confiance en son SI, l’authentification ainsi réalisée permet l’authentification mutuelle des utilisateurs (flèches rouges).

3 Conclusion

Nous avons introduit une modélisation de l’authentification dans les systèmes d’information basée d’une part sur les notions de canal et de session authentifiés, d’autre part sur la notion de mécanisme de déverrouillage. La caractéristique principale de ce modèle est que les opérations vulnérables au rejeu que sont les opérations de déverrouillage sont effectuées en local, tandis que les opérations distantes sont dévolues à des protocoles cryptographiques d’authentification entre les systèmes d’information.