Multifunctional Smartcards for Electronic Commerce - Application of the Role and Task Based Security Model

Kathrin Schier
University of Hamburg, Faculty of Informatics
D-22527 Hamburg, Germany

Abstract

Electronic commerce demands different security requirements for its many different applications. In the near future one smartcard may be used for many electronic commerce applications, such as payment systems, access to banking services and financial transactions over the Internet. A role and task based security model (R&T model) can ensure a secure access to many different services through an application based security framework. It can be used and implemented in a multifunctional smartcard in order to ensure both the users personal need for application based security and his right to informational self determination - as the fundamental right of privacy is defined in the German legal system. A successful application of the model can help the user navigate a secure way through the jungle of electronic commerce.

1 Electronic commerce

The ever increasing growth of the virtual business world will lead to more and more electronic based payment systems. The present day use of credit and debit card based payment is already widespread. It will become more commonplace, because of the increasing acceptance of such cards within the retail trade. The number of businesses accepting electronic payment is steadily growing. The introduction of the „electronic wallet“ in Germany will increase the use of electronic money and therefore decrease the use of conventional currency. In the future one or more cards will be used for a wide variety of purposes, not only in electronic commerce. In general, the probability is high that one multifunctional card will achieve wider acceptance than a large number of single function cards.

Although the Internet was never designed for secure transactions, it has become the technical medium for network based financial transactions. The need for the development of secure payment systems has therefore become blatantly apparent. The multitude of payment media, including credit and debit cards, cheques and traditional currency, has spawned a corresponding number of technical systems which support them. Even though many of them are already in use, no single system has become accepted as the de facto standard. A review of different electronic payment systems and their security aspects can be found in [1], [2] and [3].

One special property of conventional currency should not be neglected: Anonymity. Transactions cannot be traced back to individual customers, the identity of which therefore remains unknown. The billing and accounting systems related to cheques and credit cards presuppose a direct link between customer and transaction. The anonymity property cannot therefore apply to these media. Electronic payment systems need to preserve the anonymity property of traditional currency while also offering a payment mode which links individuals and their transactions. Unfortunately, most electronic wallet systems do not support anonymous use. These systems protocol individual transaction data, which allow them to reidentify individual users. The transactions are protocoled by a separate clearing house, which allows the electronic wallet transactions to be treated as a different account. Financial institutions claim that this system, by which the bank and the clearing house both store transaction data, makes checks for misuse possible and therefore more secure. The loss of anonymity may, however, affect the behaviour of the consumer and infringe on personal rights. The right to informational self determination is endangered, unless an anonymous payment system is not available.

There is a trend within the financial community towards offering services, such as money transfer, checking the status of accounts, etc., as so-called „home banking“. Home banking services can be accessed directly („online“) from the users home computer via the Internet. As homebanking becomes more and more widespread, additional services, such as stock market trading or

investment information, will also be made available. In addition to this, automatic teller machines (ATM) will eventually provide access to all banking services. Personal contact between the customer and the bank employee will be reduced and finally substituted by the networked based direct banking services. This shift in emphasis towards online banking services may not always be of advantage to the customer. These risks and other global digital commerce problems are discussed in [4].

Developments to date show that, in order to support the large number of existing electronic commerce applications, many different cards are required. A typical (physical) wallet contains a variety of cards, including bank and credit cards, social security cards, telephone cards, customer cards. This ever increasing multitude of „plastic“, with its associated volume of personal identification numbers (PIN), already tends to confuse the user. Not only does the risk of forgetting PINs increase, but also the danger of associating a card with the wrong PIN. This causes added risks, such as the customer may be tempted to write down the PINs and keep them in the wallet.

Most of these applications, such as electronic payment systems, network based financial transactions and electronic banking services, will use a smartcard for their implementation. This paper presents an electronic commerce application of the role and task based security model (R&T model) for multifunctional smartcards [5]. One multifunctional card implementation of the R&T model model could support not only several different electronic commerce applications, but also non-financial functions. For each application the user will have an individual security level, which is already wellknown from single application cards. For a specific application the user can choose the required security level, which may e.g. offer an anonymous use. Before describing the R&T model in a semi-formal way, some requirements are mentioned.

2 Requirements

For every Information Technology (IT) system security requirements have to be defined, as well as any other requirements, including functional or technical aspects. The security requirements of an IT system will be gained through a security policy, describing the security properties. A security model realises a special security policy to support the security properties of an IT system. A security model can have formal or semi-formal character. It describes the rules for accessing data objects in an IT system, enforcing e.g. the security properties of confidentiality [6] or integrity [7]. Other models, such as role based access control [8], [9], [10] or a formal privacy model [11] are very interesting for multifunctional smartcard applications. Role based access control supports access to an object according to the role in which the user is acting. The „formal privacy model“ supports access to an object depending on privacy principles associated with a users task. These two approaches can be combined to achieve individual security level for each application.

Based on role based access control model and formal privacy model, a combination of both, namely a role and task based model supports the user to perform different tasks and act in different roles. The role and task based model (R&T model) has two dimensions (roles and tasks) compared to the basic models (either roles or tasks). Therefore it supports more detailed granularity for individual security levels. The R&T Model will guarantee maximum security if it will be implemented in the security kernel of a smartcard operating system. Due to the technical conditions of a smartcard, its operating system protects the data stored in chip very efficiently.

Subjects represent user processes. A subject can choose **tasks** to perform and **roles** to act in. Tasks are hierarchically structured, so that tasks may be composed of tasks. Tasks describe what can be done, roles describe how it can be done. A subject can have one (or more) actual role-task combination. A subject can only perform more than one role-task combination, if these combinations are not mutually exclusive to each other (separation of duty). Every role-task combination provides a list of **procedures** and **objects**. The procedure-object list defines the access to objects. Objects are data objects, stored on the card.

A user chooses a task and a role in which he wants to perform the task. Within his actual role-task combination provided procedures will access defined objects. The access to objects depends on the actual role and task context. A chosen task and a chosen role are associated with a specific security. Each security level defines which data will be transmitted and what kind of information will be openly available.

The following requirements for a security policy are defined, which can be realised by the R&T model:

- The user is the **owner** of the data. While choosing individual roles and tasks he can enforce his right for informational self determination and decide himself who should access which data, stored on his smartcard. For the usage of this right it is important to have either an active user or to configure the card for the user.
- It is important to have a **configuration** for standard applications. Especially for technically inexperienced persons it should be easy to use the card without any security disadvantages. The use of the smartcard should only enable an active administration, but not enforce it.
- The **role and task based model** supports **different multifunctional applications**, not only in the area of

electronic commerce. For electronic commerce, the model should offer both an anonymous role and a non anonymous one. Another role can provide the card as an access medium for network based transaction and banking services.

- A **trustworthy institution** is necessary to support the user while administrating the card. This institution should have a consulting position for all technical, organisational and legal questions.

3 The R&T-Model

A multidimensional security model has been developed, that helps the user to use different applications with varying security requirements within the electronic commerce. The R&T model increases security and privacy. The model provides tasks for the user to perform and roles to act in. Tasks are authorised for users in a many to many relationship. Tasks can be performed in different roles. Roles are also authorised for subjects and describe a structure for action executing procedures on defined objects. Procedures accessing objects depend on the role and task context. The R&T model provides administrators with the capability to regulate who can perform what kind of actions, when and on which objects. New users can be authorised for roles very easily and can also be revoked without any effort. This concept allows a great capability of administrative support. The problem of interest conflicts between role-task combinations will be solved with the principle of static and dynamic separation of duty. This implies that a role-task combination can only be authorised for a user, if there is no other authorised combination for the user, which is mutually exclusive with the proposed combination (static separation of duty). Furthermore a user can only perform a new role-task combination, if this combination is not mutually exclusive with any other combination the user is currently performing (dynamic separation of duty).

A smartcard is suggested for implementation. It can be used as a data base and as an access medium. All personal data which is necessary for payment transactions and any other financial services is stored on the smartcard. All roles and tasks are stored on the card as well. Their definition depends on the application context. Within each role and task context, the minimum of data will only be released. The user can decide by himself which role he wants to act in and therefore which of his data is accessed. Each role and task combination defines an individual security level. To define the R&T model as a state machine model variables, rule and transition functions are needed. In this paper I concentrate on the variables and the rules. The detailed transition functions can be read in [12].

The following variables and rules describe the R&T model.

3.1 Variables

To describe the R&T model, variables are needed, such as subjects, roles, tasks, procedures and objects. A users process is represented by a subject. To perform a procedure on an object, a subject has to activate a task and be active in a role. To choose a task and a role, the task must be authorised for the subject and the role must be authorised for the task. Variables are sets of elements. To use the variables, functions are necessary to find out what elements are belonging to a special set under special conditions. The result of the function can be a single element as well as a set of elements.

Subjects: Subjects are active entities of the model. They represent the users.

$$\text{Subjects} = \{s_1,...,s_n\} \quad (1)$$

Tasks: Subjects can perform different tasks. Subjects can activate in at least one task at one time, which they can choose from a set of authorised tasks.

$$\text{Tasks} = \{t_1,...,t_m\} \quad (2)$$

Subjects can perform different tasks which are authorised for them. The function *Authorised_Tasks* shows the set of tasks, which are authorised for the subjects s_i.

$$\text{Authorised\_Tasks}(s_i) \subseteq \text{Tasks} \quad (3)$$

The function *Active_Tasks* shows the set of active tasks of the subject s_i at that time.

$$\text{Active\_Tasks}(s_i) \subseteq \text{Tasks} \quad (4)$$

Roles: A subject can act in different roles, to perform special tasks.

$$\text{Roles} = \{r_1,...,r_o\} \quad (5)$$

The roles must be authorised for the subject. The set of roles is shown by the function *Authorised_Roles.*

$$\text{Authorised\_Roles}(s_i) \subseteq \text{Roles} \quad (6)$$

The chosen active roles of a subject are given by the function *Active_Roles*.

$$\text{Active\_Roles}(s_i) \subseteq \text{Roles} \quad (7)$$

A subject can have authorised role-task combinations as well as active role-task combinations:

$$\text{Authorised\_Roles\_and\_Tasks}(s_i) \subseteq \text{Roles} \times \text{Tasks} \quad (8)$$

$$\text{Active\_Roles\_and\_Tasks}(s_i) \subseteq \text{Roles} \times \text{Tasks} \quad (9)$$

Static separation of duty: A subject can only perform more than one task or role or role-task combination, if one task (role, combination) is not mutually exclusive with any other task (role, combination) which is authorised for the user. This property preserve the policy of static separation of duty. The following functions give the set of tasks (roles, combinations) which are mutually exclusive with the proposed task t_j, (roles r_i, combinations r_i, t_j).

$$\text{Mutually\_Exclusive\_Roles}(r_j) \subseteq \text{Roles} \quad (10)$$

$$\text{Mutually\_Exclusive\_Tasks}(t_j) \subseteq \text{Tasks} \quad (11)$$

$$\text{Mutually\_Exclusive\_Roles\_and\_Tasks}(r_i, t_j) \subseteq \text{Roles} \times \text{Tasks} \quad (12)$$

Dynamic separation of duty: In some organisations a subject can be allowed to perform more than one task (role, combination) without having any conflicts of interest when acting in independently, but may entail conflicts when acting in simultaneously. Dynamic separation of duty allows more flexibility in simultaneous performing of tasks (roles, combinations). The following functions show the list of mutually exclusive tasks (roles, combinations) for a proposed active task t_j, (role r_i, combination r_i, t_j).

$$\text{Mutually\_Exclusive\_Active\_Roles}(r_i) \subseteq \text{Roles} \quad (13)$$

$$\text{Mutually\_Exclusive\_Active\_Tasks}(t_j) \subseteq \text{Tasks} \quad (14)$$

$$\text{Mutually\_Exclusive\_Active\_Roles\_and\_Tasks}(ri, tj) \subseteq \text{Roles} \times \text{Tasks} \quad (15)$$

Procedures: Within each task and role combination procedures are provided to access objects in a controlled way. Procedures realise the controlled access to objects. There is a set of procedures which can be used in the model.

$$\text{Procedures} = \{p_1,...,p_p\} \quad (16)$$

Objects: Objects are passive entities of the model, the data objects.

$$\text{Objects} = \{o_1,...,o_r\} \quad (17)$$

Allowed Access: The controlled access to objects are realised with procedures depending on the role-task combination of a subject. In the beginning it has to be defined, which procedures should access which objects in which role-task-context. The set of allowed access is specified by a tupel with the elements subject, role, task and a list of several pairs, containing procedures and objects.

$$\text{Allowed\_Access} = \{(s_i, r_j, t_k, [(p_l, o_m)])\} \quad (18)$$

The function *Active_Access* shows the access to one or more objects, which is allowed at that time within the proposed framework of the subjects chosen role and task. The result of the function gives a list of pairs (procedure, object) with at least one element.

$$\text{Active\_Access}(s_i, r_j, t_k) = [(p_l, o_m)] \quad (19)$$

3.2 Rules

The following rules define conditions for the R&T model, which have to be fulfilled to meet the system requirements, mentioned above.

Rule 1: Role Authorisation

A task can never be performed in an active role if it this role not authorised for that task.

$$\forall \text{ r:Roles: } \text{Active\_Roles}(t) \in \text{Authorised\_Roles}(t) \quad (20)$$

Rule 2: Task Authorisation

A subject can perform a task, only if the task belongs to the set of authorised tasks for the subject.

$$\forall \text{ s:Subjects: } \text{Active\_Tasks}(s) \subseteq \text{Authorised\_Tasks}(s) \quad (21)$$

Rule 3: Role and Task Authorisation

A subject can perform a role-task combination, only if the role-task combination belongs to the set of authorised combinations for the subject.

$\forall$ s:Subjects:
Active_Roles_and_Tasks (s) $\subseteq$
Authorised_Roles_and_Tasks (s) (22)

Rule 3: Task Hierarchy

Tasks can be structured hierarchically. Tasks can be represented as a parent relationship, as an ordered pair $((t_{i+1}, t_i), >)$, where t_{i+1} is the parent and t_i is the child. The relation „>“ means „contains“. If a task t_j is authorised for a subject s and contains the task t_i, then the task t_i is also authorised for the subject s.

$\forall$ s:Subjects, $t_{i,j}$:Tasks, $i \neq j$:
$t_j \in$ Authorised_Tasks (s) $\wedge$ $t_j > t_i \Rightarrow$
$t_i \in$ Authorised_Tasks (s) (23)

Rule 5: Static Separation of Duty

5.1: A role can be authorised for a subject, only if there is no other authorised role for the subject, that is mutually exclusive with the proposed role.

$\forall$ s:Subjects, $r_{i,j}$:Roles, $i \neq j$:
$r_i \in$ Authorised_Roles (s) $\wedge$
$r_j \in$ Authorised_Roles (s) $\Rightarrow$
$r_i \notin$ Mutually_Exclusive_Roles (r_j) (24)

5.2: A task can be authorised for a subject, only if there is no other authorised task for the subject, that is mutually exclusive with the proposed task.

$\forall$ s:Subjects, $t_{i,j}$:Tasks, $i \neq j$:
$t_i \in$ Authorised_Tasks (s) $\wedge$
$t_j \in$ Authorised_Tasks (s) $\Rightarrow$
$t_i \notin$ Mutually_Exclusive_Tasks (t_j) (25)

5.3: A role-task combination can be authorised for a subject, only if there is no other authorised role-task combination for the subject, that is mutually exclusive with the proposed combination.

$\forall$ s:Subjects, $r_{i,j}$:Roles, $t_{k,l}$:Tasks, $(r_i, t_k) \neq (r_j, t_l)$:
$(r_i, t_k) \in$ Authorised_Roles_and_Tasks (s) $\wedge$
$(r_j, t_l) \in$ Authorised_Roles_and_Tasks (s) $\Rightarrow$
$(r_i, t_k) \notin$ Mutually_Exclusive_Roles_and_Tasks (r_j, t_l) (26)

Rule 6: Dynamic Separation of Duty

6.1: A subject can perform a new role, only if this role is not mutually exclusive with any other roles the subject is currently acting in. Two or more active roles of one subject at the same time must not be mutually exclusive.

$\forall$ s:Subjects, $r_{i,j}$:Roles, $i \neq j$:
$r_i \in$ Active_Roles (s) $\wedge$ $r_j \in$ Active_Roles (s) $\Rightarrow$
$r_i \notin$ Mutually_Exclusive_Active_Roles (r_j) (27)

6.2: A subject can perform a new task, only if this task is not mutually exclusive with any other tasks the subject is currently performing. Two or more active tasks of one subject at the same time must not be mutually exclusive.

$\forall$ s:Subjects, $t_{i,j}$:Tasks, $i \neq j$:
$t_i \in$ Active_Tasks (s) $\wedge$ $t_j \in$ Active_Tasks (s) $\Rightarrow$
$t_i \notin$ Mutually_Exclusive_Active_Tasks (t_j) (28)

6.3: A subject can perform a new role-task combination, only if this role-task combination is not mutually exclusive with any other combination the subject is currently performing.

$\forall$ s:Subjects, $r_{i,j}$:Roles, $t_{k,l}$:Tasks, $(r_i, t_k) \neq (r_j, t_l)$:
$(r_i, t_k) \in$ Active_Roles_and_Tasks (s) $\wedge$
$(r_j, t_l) \in$ Active_Roles_and_Tasks (s) $\Rightarrow$
$(r_i, t_k) \notin$ Mutually_Exclusive_Active_Roles_and_Tasks (r_j, t_l) (29)

Rule 7: Object Access

While planning an application it has to be specified, which procedures should have access to which objects depending on the role-task-context a subject has. A procedure can be executed only if there is a role and task context for the proposed subject, which allows the specified access to objects via procedures.

$\forall$ s_i: Subjects, r_j: Roles, t_k: Tasks, p_l: Procedures,
o_m: Objects,:
$([(p_l, o_m)]) \in$ Active_Access $\Rightarrow$
$(r_j, t_k) \in$ Active_Roles_and_Tasks(s) $\wedge$
$(s_i, r_j, t_k, [(p_l, o_m)]) \in$ Allowed_Access (30)

Beside variables and rules, transition functions are necessary to describe a state machine model. These are the following:

- Choose task
- Choose role
- Execute

It is possible for the user to change the order in choosing. There are two possibilities: Choosing first a task and then a role, or choosing first a role and then a task. It makes no difference for the execution, so that two ways are allowed, see figure 6.

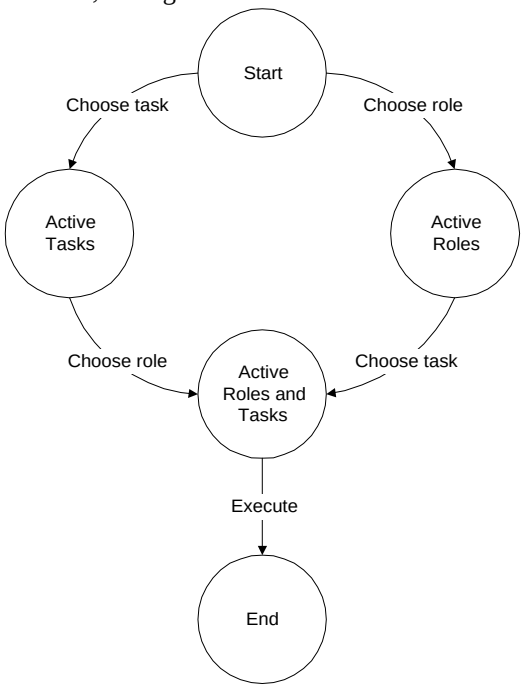

figure 1: Transition functions

4 Application

In order to show the applicability of the R&T model to the area of electronic commerce the following example with hypothetical users, tasks and roles has been chosen. It is a german application example, with regard to common german electronic commerce activities.

In this application example the card is used for different purposes within the electronic commerce. The first decision is WHAT should be used, this leads to the task. The second decision is HOW should the task be performed, now it comes to the role. The following symbols are used to specify a *card application*, with its *tasks* and *child task* as well as their *roles* a *user* can act in.

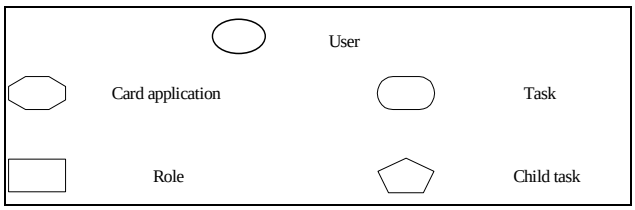

figure 2: Used symbols

The variables in this example are the following:

Subjects = {card holder, bank},
Tasks = {Money transfer, Banking, Administrating},
Roles = {Electr. wallet holder, EC-card holder, Credit card holder, Bank administrator},
Procedures = {Read, Write, Append, Delete, Create},
Objects = {Name, Account no, Bank information, Creditcard no, Limits}.

The application has a *money transfer* task, a *banking* task and an *administrating* task. Possible roles are Electronic wallet holder, EC-card holder, Credit card holder and Bank administrator. The use of a smartcard is assumed, whereby the *card holder* and a *bank* constitute the set of permissible card users, shown in figure 2.

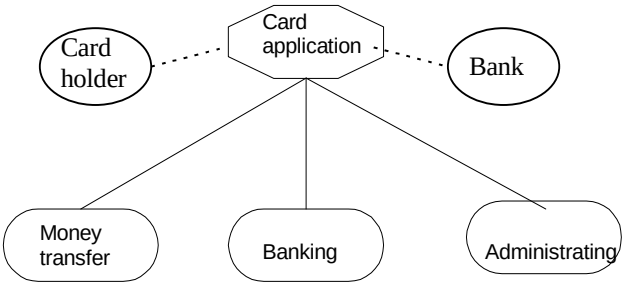

figure 3: Application and tasks

The application of the card is parted into an electronic commerce use, containing money transfer and banking and an administrating use. The three tasks of *money transfer*, *banking* and *administrating* are authorised tasks for the card holder. For the bank only the task of administrating is authorised.

Authorised_Tasks(Cardholder) = {Money transfer, Banking, Administrating}

Authorised_Tasks(Bank) = {Administrating}

The authorised roles for the subjects are the following:

Authorised_Roles(Cardholder) = { Electronic wallet holder, EC-card holder, Credit card holder}

Authorised_Roles(Bank) = {Bank administrator}

The tasks can be composed of tasks. In the following the three tasks are described in detail. The task of money transfer and its child tasks is shown in figure 3. The money transfer task contains two tasks within this hierarchy. The *money transfer* can be money *paying* or money *accepting*. These two tasks can be done in different roles describing HOW the tasks are performed. Both the paying and the accepting task can be performed in the role of an *electronic wallet holder*, in the role of an *EC-card holder* or in the role of a *credit card holder*. The

requirements for the task hierarchy are mentioned in rule 4, chapter 3.

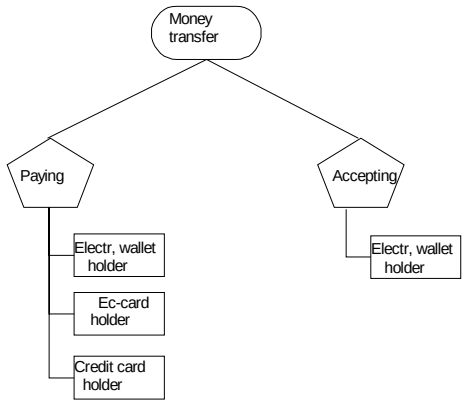

figure 4: Money transfer task and its roles

A task hierarchy is not necessary for each application. Task hierarchy supports a detailed composition of tasks and duties and therefore allows a high granularity. On the lowest task level (the child task) different roles are provided.

Figure 4 shows the task of *banking* providing all typical banking functions. This task contains child tasks as well as shown in the money transfer task. A user performing the banking task can *load* his *electronic wallet* (child task), can use *money transfer* (child task), can get *account statements* (child task) and other *accounting functions* (child task).

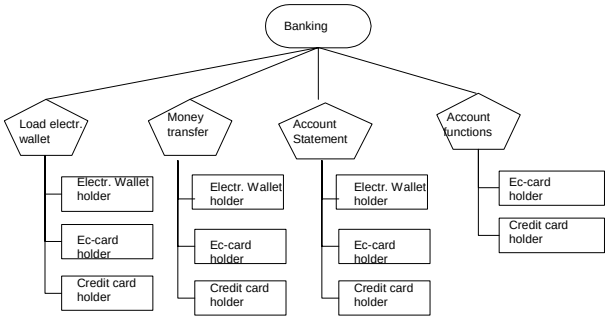

figure 5: Banking task and its roles

The task *load* money on the *electronic wallet* can be done in the role of the *electronic wallet holder*, which means loading cash money on the card. In the role of a *EC-card holder* the loaded money will be transferred from the EC-account of the user. If the role of the *credit card holder* is used the money loaded on the electronic wallet will be credited from the credit card account.

The user, who has chosen the task of *money transfer* can also choose the role in which he wants to perform the task of money transfer. In dependency of the chosen role, the money will be credited from the electronic wallet, from the EC-account or from the credit card account.

The task *account statement* is also possible in all roles. Dependent from the role the statement of the electronic wallet, of the EC-account or of the credit account will be listed. Only the task *account functions* is allowed for only two roles, the EC-card and the credit card holder.

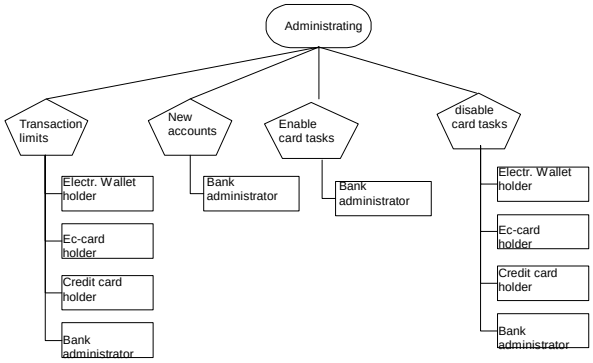

figure 6: Administrating task and its roles

The *administrating* task is parted into card configuration and management, see figure 5. In this example there are four administrating child tasks, *transaction limits*, *new accounts*, *enable card tasks* and *disable card tasks*. All of these four can be performed in the role of the *bank administrator*. He can set new transaction limits can create new accounts or can enable additional card tasks or disable old card tasks. In the roles of the *electronic wallet holder*, the *EC-card holder* and the *credit card holder* only two of the four tasks are allowed. The task of *transaction limits* can be performed in these three roles, because it makes sense if the special card holder wants to reduce his transaction limits in order to make sure not to spend too much money. These three roles are also allowed to *disable* some *card tasks*, if there are not wanted to be used.

For using the electronic wallet card, the user can choose a task to perform and a role to act in. After choosing a role and a task, the variables of the active roles and tasks are set. If the user wants to perform more than one role-task combination at the same time, the rule of dynamic separation of duty is checked. If the desired combinations are not mutually exclusive, they can be performed at the same time without any problems.

5 Conclusion

The role and task based security model is a framework which guarantees the individual security needs of its user. It provides many mechanisms which allow both secure

and selective access to user data. The semi-formal model described above has been adapted to the special needs of electronic commerce. Roles and tasks have been defined, which provide maximum individual security, while allowing detailed access to electronic commerce services. An ordered list of procedures and objects must be specified for each role and task combination. This defines the permissible access to individual and public data. Procedures use and access different data objects, depending on the role and task combination.

The R&T model is not limited to applications within electronic commerce, but may also be applied to many other IT associated areas, where individual security requirements for different applications must be considered. The implementation of the R&T model on a smartcard, combined with biometric identification and enhanced authenticity, may facilitate both a secure and comfortable use of a wide variety of applications. The implementation is not limited to smartcards. It can also be realised complex computer systems without using smartcards, for example health care application systems.

The acceptance of R&T modelled electronic commerce systems is necessary and trust into these systems has to be build up. At the moment it is not clear if the shift away from traditional commerce and towards electronic commerce presents a chance or a risk to society in general.

References

[1] K. Schier: Vergleich und Bewertung aktueller Systeme im elektronischen Zahlungsverkehr, *Proceedings of the German UNIX User Group Conference (GUUG)*, September 1997.

[2] A. Engel, A. Lessig, K. Schier: Chipkartenbasierte Zahlungssysteme - Der Große Bruder im Portemonnaie, in: *Proceedings of the OmniCard '98*, Berlin, January 1998, p. 19-43

[3] K. Schier: Zahlungssysteme im Internet - Eine sicherheitstechnische Bewertung, *Proceedings of the Deutsche Internet Kongress (DIK)*, May 1998.

[4] K. Brunnstein, K. Schier: Global Digital Commerce: Impacts and Risks for Developments of Global Information Societies", in: J.Berleur and Diane Whitehouse, Ed., 'An ethical global information society: culture and democracy revisited', Proceedings of the IFIP WG 9.2 Corfu international conference, 8.-10. May 1997, Chapman & Hall

[5] K. Schier: A Role and Task Based Security Model for Multifunctional Smartcard Applications in the Area of Electronic Commerce, *Proceedings of the IFIP WCC '98, SEC'98 Conference*, Wien, Budapest, Kluwer, September 1998.

[6] D. E. Bell, L. J. LaPadula: Secure Computer Systems: Mathematical Foundations, *Technical Report, ESD-TR-73-278, Volume 1*, The MITRE Corporation, Bedford, MA, March 1973.

[7] D. Clark, D. R. Wilson: A Comparison of Commercial and Military Computer Security Policies, *Proceedings of the IEEE Symposium on Computer Security and Privacy*, April 1987.

[8] D. Ferraiolo, R. Kuhn: Role-Based Access Controls, *Proceedings of the 15th National Computer Security Conference*, Baltimore MD, October 1992.

[9] D. Ferraiolo, J. A. Cugini, R. Kuhn: Role-Based Access Control (RBAC): Features and Motivation, *Eleventh Annual Computer Security Applications Conference ACSAC ´95*, New Orleans, Louisiana, 11-15 December 1995, *IEEE Computer Society Press*, Los Alamitos, December 1995.

[10] R. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman: Role Based Access Control Models, *IEEE Computer, 29(2)*, February 1996.

[11] S. Fischer-Hübner: Considering Privacy as a Security-Aspect: A Formal Privacy-Model, DASY Paper No 5/95, Institute of Computer and Systems Sciences, Copenhagen Business School, Copenhagen, May 1995.

[12] K. Schier: Vertrauenswürdige Kommunikation im elektronischen Zahlungsverkehr - Ein rollen- und aufgabenbasiertes Sicherheitsmodell für multifunktionale Chipkartenanwendungen, Dissertation am Fachbereich Informatik, 1999.