HAL
archives-ouvertes.fr

Développement combiné et prouvé de systèmes transactionnels cryptologiques

Nazim Benaissa, Dominique Méry

▶ To cite this version:

Nazim Benaissa, Dominique Méry. Développement combiné et prouvé de systèmes transactionnels cryptologiques. Approches Formelles dans l'Assistance au Développement de Logiciels - AFADL 2009, Jan 2009, Toulouse, France. 2009. <inria-00426405>

HAL Id: inria-00426405
https://hal.inria.fr/inria-00426405

Submitted on 26 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Développement combiné et prouvé de systèmes transactionnels cryptologiques

Nazim Benaïssa[1,2,4] and Dominique Méry[1,3,4⋆]

[1] Université Henri Poincaré Nancy 1
[2] benaissa@loria.fr
[3] mery@loria.fr
[4] LORIA
BP 239
54506 Vandœuvre-lè-Nancy
France

Abstract. Dans le cadre du développement prouvé et incrémental de systèmes informatiques, nous nous intéressons à des systèmes transactionnels présents dans les échanges bancaires comme Mondex. Nous étudions la combinaison de modèles Event B comme un modèle du Mondex et un modèle de protocoles cryptologiques afin d'obtenir un système plus robuste aux attaques. Puis, nous montrons sur quels principes est fondée cette combinaison et nous montrons dans quelles mesures elle peut simplifier la tâche de modélisation prouvée et incrémentale.

1 Introduction

Le dévelopement de systèmes informatiques repose sur des techniques mises en œuvre dans le cadre d'outils et de méthodologies. Dans le cas de systèmes critiques, ce dévelopement nécessite d'utiliser des méthodes fondées sur des techniques formelles et aussi d'intégrer le savoir-faire pratique en produisant des études de cas pertinentes montrant l'intérêt de ces approches. Nous nous plaçons dans le cadre de systèmes transactionnels mis en œuvre dans des smartcards et nous considérons des opérations de développement permettant d'enrichir ou de superposer plusieurs modèles développés. Nous utilisons le cas d'étude du Mondex proposé dans le cadre du Grand Challenge[8] et constituant un système transactionnel relativement complexe pour être illustratif de notre technique; puis, nous utilisons un modèle de protocole cryptographique développé aussi de manière incrémentale et prouvée. Cette combinaison ou composition de modèles développés prouvés est générale et s'appuie sur le raffinement. On obtient ainsi un système transactionnel intégrant un protocole cryptographique en conformité avec le modèle d'attaques de Dolev et Yao [6, 3]. La contribution de ce document est donc un patron de conception prouvé qui s'appuie sur le raffinement et qui est applicable à des systèmes transactionnels.

⋆ Ce travail a été fait dans la cadre du projet RIMEL No. ANR-06-SETI-015-03

Grandy et ses coauteurs [7] apportent un traitement de la vérification du Mondex en intégrant les éléments propres à la cryptographie; leur approche est fondée sur une hypothèse de choix de la méthode cryptologique retenue et sur une preuve à l'aide de l'outil KIV. Cet exercice nous est apparu comme intéressant et nous avons donc estimé qu'il serait plus opportun d'utiliser une approche produisant un système Mondex intégrant des éléments cryptologiques et correct par construction. De ce fait, une telle construction n'a pas encore été réalisée. Cependant, cette idée s'appuie sur le développement de protocoles de transport intégrant des éléments cryptologiques. Dans un autre travail, nous avons développé un certain nombre de protocoles cryptologiques qui intégraient le modèle d'attaque de Dolev et Yao et qui étaient développés de manière incrémentale et prouvée en Event B. De ce point de vue-là aussi, si les protocoles ne sont pas nouveaux, leur développement est nouveau selon cette technique. Ce travail n'a donc pas de réel concurrent dans la mesure où d'autres techniques plus directes comme Isabelle [12, 11] pourraient être utilisées mais sans la dimension incrémentale de Event B. Cet exercice étend aussi le spectre de Mondex et enrichit les travaux sur ce type de systèmes en ajoutant une dimension cryptologique qui est loin d'être simple mais qui trouve un traitement relativement maîtrisé dans ce travail. Aussi, notre choix de développer Mondex [13, 1] s'appuie sur la complexité inhérente du cas d'étude et sur l'intérêt de cette étude dans le cadre du Grand Challenge [8]. Michael Butler [5] a développé ce système en B événementiel sans la dimension cryptologique. Ce travail va au-delà du simple développement d'une étude de cas et apporte une illustration de la manière de combiner des modèles formels de système.

La suite du document est organisée comme suit. La section suivante 2 rappelle ce qu'est Mondex et donne son développement en Event B. La section 3 apporte des éléments sur la manière de développer les modèles du protocole cryptologique utilisé dans notre développement combiné. Puis, dans la section 4, nous donnons une description du mécanisme de composition utilisé fondé sur le raffinement et nous montrons comment on peut poursuivre le développement par raffinement du modèle résultant de la composition. Enfin, nous concluons dans la section 5 sur cette technique de composition.

2 Présentation de Mondex

Mondex [13, 1] est un système de paiement électronique qui fut introduit pour la première fois par la *National Westminster Bank* en 1990 avant d'être repris par *MasterCard*. Le système est basé sur les cartes à puce servant de porte-monnaie électroniques contenant chacun un solde d'argent. Une des spécificités du système est que les transactions sont *hors-ligne*, ceci signifie que les transactions se passent entre deux porte-monnaie sans l'intervention d'une autorité centralisée. Au cours d'une transaction, chaque porte-monnaie doit donc être capable de garantir lui-même les mesures de sécurité inhérente à ce genre de système, sans l'intervention d'une quelconque autorité de régulation.

Le protocole *Mondex* Chaque porte-monnaie est initialement crédité d'un solde. La transaction consiste en un transfert d'argent entre un porte-monnaie source et un porte-monnaie destination. La figure 1 montre le schéma global d'une transaction.

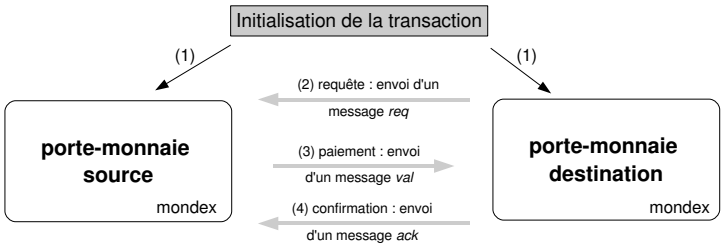

Fig. 1. Les différentes étapes d'une transaction

Le schéma de la figure 1 montre le cas d'une transaction où tout se passe bien. Mais dans la réalité plusieurs événements peuvent perturber le déroulement d'une transaction. Il peut s'agir par exemple d'une défaillance du canal de communication ou encore d'un retrait prématuré de la carte à puce du lecteur de carte, avant que la transaction n'arrive à son terme. Pour prévenir ce genre d'erreurs, un certain nombre de dispositifs ont été prévus. En plus de son identifiant et de son solde, un porte-monnaie contient un journal d'erreurs dans lequel les transactions qui ont échoué sont transcrites. Il contient également le numéro de séquence de la prochaine transaction à effectuer. Un porte-monnaie a aussi un état qui varie au fur et à mesure que la transaction avance. Les différents états possibles sont *eaFrom, eaTo, epr, epv, epa* (voir figure 2).

Lorsqu'un porte-monnaie source abandonne la transaction prématurément alors qu'il a déjà débité son solde (*état epa*), il inscrit la transaction en cours sur son propre journal d'erreurs. De même pour un porte-monnaie cible qui a envoyé un message de type *Req* et qui n'a pas encore crédité son solde (*état epv*). Grâce à ce procédé, le croisement des informations contenues dans les journaux d'erreurs des différents porte-monnaie permettra de récupérer l'argent apparemment perdu localement.

2.1 Développement du Mondex

Le développement du Mondex a été réalisé en plusieurs étapes successives : un premier modèle abstrait suivi de dix raffinements qui rendront le modèle de plus en plus concret jusqu'à ce que tous les aspects du protocole soient pris en

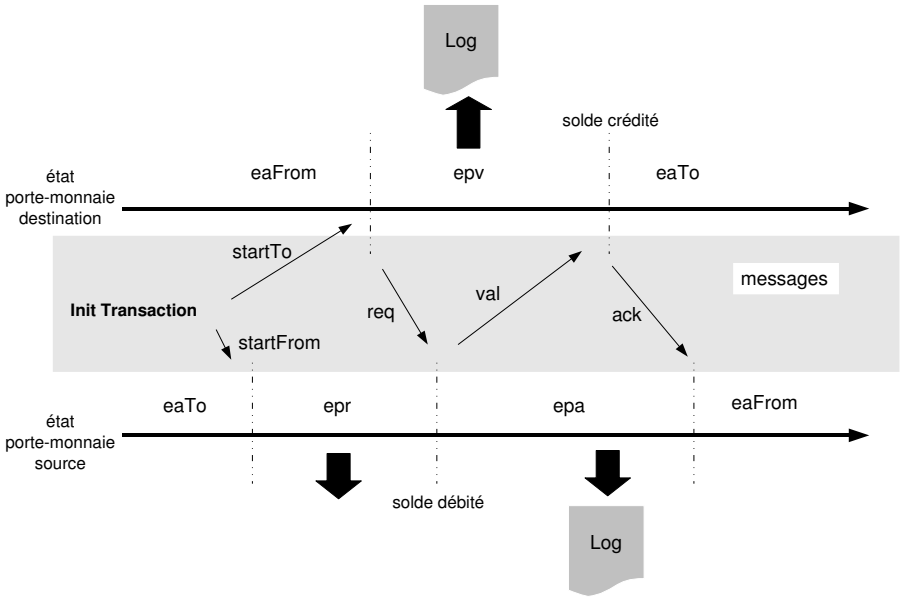

Fig. 2. Le protocole Mondex

considération.
Dans le premier modèle abstrait, la transaction entre deux porte-monnaie est atomique, le transfert d'argent se fait en un coup. Le solde du porte-monnaie source est débité au même temps que le solde du porte-monnaie cible est crédité (figure 3). Dans ce premier modèle sont exprimées les principales propriétés de

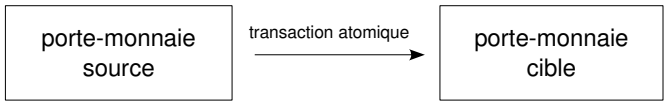

Fig. 3. Une transaction dans le modèle abstrait

sûreté que le système doit garantir, à savoir:

- Pas d'argent créé dans le système.
- Pas d'argent perdu dans le système.

Modèle abstrait : Transfert atomique Dans ce premier modèle, les porte-monnaie ont un identifiant propre à chacun. Tous les identifiants possibles sont

contenus dans un ensemble $NAME$. Chaque porte-monnaie contient en plus de son nom deux paramètres :

- Solde courant du porte-monnaie.
- Argent perdu localement : quand une transaction échoue, l'argent qui n'est pas arrivé au porte-monnaie cible est stocké comme argent perdu localement sur le porte-monnaie source.

Un certain nombre de types sont introduits pour les besoins de la modélisation :

SETS

$NAME$ /* ensemble des identifiants de porte-monnaie */

L'état du système à un moment donné est reflété par des variables qui contiennent le solde et l'argent perdu localement par chaque porte-monnaie :

VARIABLES

$abAuthB$ /* solde de chaque porte-monnaie authentique */
$abAuthB'$ /* sauvegarde de l'ancienne valeur de la variable abAuthB */
$abAuthL$ /*argent perdu localement par chaque porte-monnaie */
$abAuthL'$ /* sauvegarde de l'ancienne valeur de la variable abAuthL */

INVARIANTS

$inv1 : abAuthB \in NAME \rightarrow \mathbb{N}$
$inv2 : abAuthB1 \in NAME \rightarrow \mathbb{N}$
$inv3 : abAuthL \in NAME \rightarrow \mathbb{N}$
$inv4 : abAuthL1 \in NAME \rightarrow \mathbb{N}$

L'invariant quant à lui exprime les deux propriétés de sûreté énoncées précédemment, pas d'argent créé frauduleusement(SUM est une fonction calculant la somme des soldes contenus dans tous les porte-monnaie).

$$inv5 : SUM(abAuthB) \leq SUM(abAuthB')$$

Par ailleurs, l'argent ne peut être perdu globalement, ceci est exprimé par le fait que la somme des soldes et de l'argent perdu localement par tous les porte-monnaie demeurent inchangés:

$$inv6 : SUM(abAuthB) + SUM(abAuthL) = SUM(abAuthB') + SUM(abAuthL')$$

Les transactions étant atomiques dans ce premier modèle, il y a donc que deux événements $TransfertOk$ et $TransfertLost$. En un coup la transaction est soit réussie ou non et les variables sont modifiées en conséquence:

EVENT TransfertOk
ANY
A
B
$money$
WHERE
$grd1 : A \in NAME \wedge B \in NAME \wedge money \in \mathbb{N}$
$grd2 : abAuthB(A) \geq money$
$grd3 : A \neq B$
THEN
$act1 : abAuthB1 := abAuthB$
$act2 : abAuthL1 := abAuthL$
$act3 : abAuthB := abAuthB \lhd\!\!\!- \{A \mapsto (abAuthB(A) - money), B \mapsto (abAuthB(B) + money)\}$
END

Contrairement à l'événement $TransfertOk$, dans l'événement $TransfertLost$ l'argent n'est pas transféré vers le solde du porte-monnaie destination mais il est stocké comme argent perdu localement par le porte-monnaie source dans la variable $abAuthL$.

Les raffinements Nous donnons en ce qui suit un aperçu du développement du Mondex nécessaire à la compréhension de ce papier. Le développement complet peut être trouvé dans [2]. Pour implémenter le protocole Mondex (voir figure 2), chaque porte-monnaie électronique a une structure complexe comme le montre la figure 4.

Il est donc nécessaire d'introduire les détails du protocole étape par étape pour réduire l'effort de preuve. Nous avons utilisé pour cela la notion de transaction qui est très abstraite dans un premier raffinement avant d'arriver aux transactions sous leur forme concrète de Mondex au bout du dixième raffinement.

Deux nouveaux ensembles porteurs sont introduits, T qui contient l'ensemble des toutes les transactions possibles. Chaque transaction a un porte-monnaie source, un porte-monnaie destination et une valeur. Des sous-ensembles de T contiennent les transactions selon leur progression comme le montre la figure 2, il s'agit des ensembles $eaFrom, eaTo, epr, epa, epv$. Une transaction va donc passer d'un ensemble à un autre durant son exécution. La variable $conBalance$ modélise le solde concret d'un porte-monnaie.

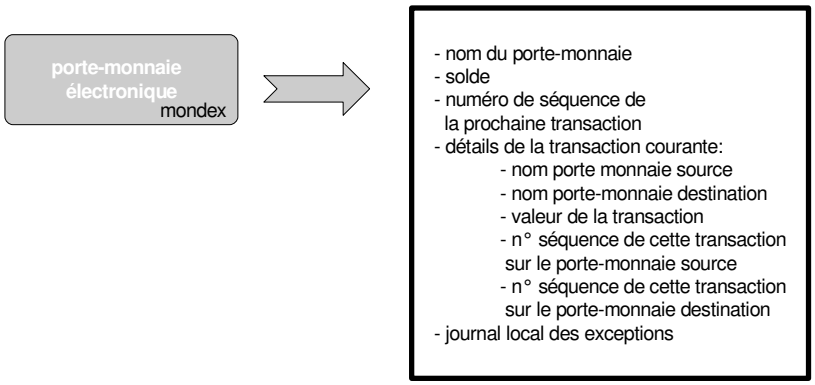

Fig. 4. La structure d'un porte-monnaie électronique

SETS

T

VARIABLES

t_src
t_dst
t_value
epv
epa

INVARIANT

$inv7 : t\_src \in T \rightarrow NAME$
$inv8 : t\_dst \in T \rightarrow NAME$
$inv9 : t\_value \in T \rightarrow \mathbb{N}$
$inv10 : epv \subseteq T$
$inv11 : epa \subseteq T$

END

Les événements du modèle représentent les différentes étapes du protocole, l'événement suivant modélise l'étape où le protocole source débite son solde: le porte-monnaie source déduit la somme *money* de son solde alors que la transaction passe de l'étape *epr* à *epv*. Il faut noter que nous avons délibérément omis beaucoup de détails sur le développement du système Mondex en ne présentant

que ce qui est nécessaire à la compréhension de ce travail.

EVENT Event_DEB
ANY
t
A
B
$money$
WHERE
$grd1 : t \in epr$
$grd2 : A = t\_src(t)$
$grd3 : B = t\_dst(t)$
$grd4 : money \in \mathbb{N}$
THEN
$act1 : epr := epr \setminus \{t\}$
$act2 : epa := epa \cup \{t\}$
$act3 : conBalance(A) := conBalance(A) - money$
END

Le reste des dix raffinements introduit pas à pas les différents détails du protocole jusqu'à arriver au protocole complet tel montré dans la figure 2. Ce modèle satisfait par construction les deux propriétés de sûreté exprimées dans le modèle abstrait qui garantissent que l'argent n'est jamais perdu. Mais ces propriétés ne garantissent pas que l'argent aille dans le bon porte-monnaie. En effet un intrus malveillant pourrait envoyer des messages avec une fausse identité et recevoir l'argent à la place d'un autre porte-monnaie. D'où le besoin d'introduire une nouvelle propriété de sécurité : l'authentification. Cette propriété signifie qu'un porte-monnaie est sûr de l'identité du porte-monnaie avec lequel il est engagé dans une transaction. Pour satisfaire cette propriété nous utiliserons des protocoles cryptographiques, plus précisément une classe de ces protocoles appelés *protocoles de transport de clés*. Nous avons dans un travail séparé développé un patron de développement permettant de modéliser des protocoles de cette classe et prouvé leur fiabilité. D'où l'idée de combiner deux modèles développés séparément, d'un côté le Mondex et de l'autre les protocoles de transport de clés afin d'obtenir un modèle héritant des propriétés des deux modèles originaux. Il faut noter que les deux modèles ont été développés à l'origine de manière totalement indépendante sans penser à les combiner par la suite. Avant de détailler les différents aspects liés à cette combinaison, nous introduisons dans ce qui suit brièvement notre patron de développement des protocoles de transport de clé.

3 Les protocoles de transport de clé

Lorsque deux agents A et B d'un système ont besoin de communiquer de manière sûr, ils ont besoin d'une clé cryptographique qu'ils sont les seuls à connaitre.

Cette clé est appelée une clé de session. Une fois cette clé de session établie, les deux agents peuvent communiquer de manière sûre. Différents protocoles permettent à deux agents d'obtenir une clé de session, parmi eux les protocoles de transport de clé où un agent propose à l'autre la clé à utiliser. Ces protocoles doivent satisfaire un ensemble de propriétés. Les deux propriétés les plus importantes pour ce type de protocoles sont :

- A la fin du protocole la nouvelle clé de session générée n'est connue que par A et B.
- A et B ne doivent pas utiliser des clé établies lors de sessions précédentes.

Un exemple de protocole de transport de clé est le protocoles de *Blake-Wilson-Menezes key transport protocol* [4] présenté dans la figure 5. Ce protocole est basé sur l'utilisation des clés publiques des agents A et B. Dans ce protocole l'agent B crée une nouvelle clé de session K_{BA} et l'envoie à l'agent A. Ce protocole est basé sur l'utilisation des signatures avec des clés publiques pour obtenir une authentification mutuelle entre A et B.

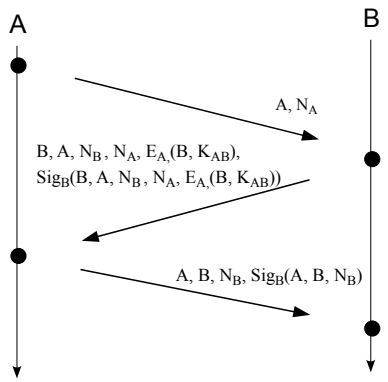

Fig. 5. Blake-Wilson-Menezes key transport protocol.

3.1 Le modèle de l'attaquant

Nous avons adopté dans notre travail un attaquant agissant selon le modèle Dolev-Yao [6]. Dans ce modèle l'attaquant a le contrôle total du canal de communication :

- Il peut intercepter et effacer n'importe quel message.
- Il peut générer un nombre infini de messages.
- Il peut décrypter des parties d'un message s'il possède la clé appropriée.
- Il peut décomposer les messages non cryptés.

3.2 Le patron de développement

Notre modèle modélisant les protocoles de transport de clé est basé sur des transactions d'abord abstraites dans le premier modèle et qui sont concrétisées plus tard notamment avec l'introduction des nonces des protocoles cryptographiques. Un nonce est un message aléatoire créé par l'un des participants. Une description complète du modèle peut être trouvée dans [2]. Ce travail est la continuité de travaux précédents concernant la modélisation d'un attaquant suivant le modèle de Dolev-Yao qui peut être trouvé dans [3]. Un bref aperçu des modèles est donné en ce qui suit:

- Le modèle abstrait: Dans ce premier modèle les différentes étapes du protocole sont modélisées avec la notion de transactions abstraites. Une transaction a plusieurs attributs tels que sa source et sa destination. Ces attributs sont utilisés pour exprimer les différentes propriétés de sûreté de notre modèle.
- Premier raffinement: Dans ce premier raffinement, nous avons introduit le reste des détail du protocole (notamment la structure des messages) qui n'ont pas déjà été introduits dans le premier modèle pour rendre le premier modèle abstrait plus générique et commun aux différents protocoles. L'événement associé à l'attaquant maintient l'invariant préservé.
- Second raffinement: Dans le second raffinement, l'événement associé à l'attaquant modélise un comportement de l'attaquant du style Dolev-Yao. Les connaissances de l'attaquant sont modélisées et elles sont utilisées pour prouver que l'attaquant n'arrive pas à violer les propriétés de sûreté.
- Troisième raffinement: Le troisième raffinement est un raffinement de données où les transactions abstraites sont remplacées par des nonces concrets.

Nous avons utilisé un ensemble porteur T contenant l'ensemble des transactions, ainsi que des attributs semblables à ceux utilisés pour le Mondex tels que la source et la destination d'une transaction. Une transaction passe cette fois par deux étapes, d'abord *progress* pour les transactions en cours et *auth* une fois l'authentification achevée et la clé de session obtenue. Nous avons également introduit un attribut t_bld_dst. En effet il faut distinguer entre la destination avec laquelle un agent croit communiquer t_bld_dst et la vraie destination (qui peut être en réalité l'attaquant) contenue elle dans t_dst. L'invariant consiste a démontrer que ces deux variables coïncident pour les transactions terminées :

$$inv12 : \forall t \cdot t \in auth \Rightarrow t\_dst(t) = t\_bld\_dst(t)$$

Les clés de sessions sont introduites dans le deuxième raffinement, lorsqu'une transaction arrive à son terme elle se voit associer une clé de session.

SETS

P_KEY

$$inv13 : t\_key \in progress \rightarrowtail P\_KEY$$

Il faut noter que la fonction associant une clé de session à chaque transaction est injective afin de garantir aux agents l'utilisation d'une nouvelle clé à chaque transaction. Pour prouver qu'une clé n'est connue que par les deux agents impliqués dans la transaction correspondante, une variable K_Mem a été introduite. Cette variable contient la mémoire de chaque agent par rapport aux clés qu'il connaît.

$$inv14 : K\_Mem \in Agent \rightarrow \mathbb{P}(P\_KEY)$$

Un invariant a été ajouté pour exprimer le fait qu'une clé n'est connue que par les deux agents impliqués dans la transaction correspondante, I correspond à l'intrus :

INVARIANTS

$$inv15 : \forall t, K \cdot t \in progress \wedge K = t\_key(t) \wedge K \in K\_Mem(I) \Rightarrow (I = t\_bld\_dst(t) \vee I = t\_src(t))$$

Nous avons appliqué ce patron de développement sur différentes études de cas notamment avec les protocoles de *Blake-Wilson-Menezes key transport protocol* [4] et de *Needham-Schroeder public key protocol* [10].

4 La combinaisons des modèles

Les deux modèles, que ce soit le Mondex ou les protocoles cryptographiques sont basés sur la notion de transaction abstraite. Certains attributs de ces transactions sont partagés par les deux modèles tels que t_src. Il faut noter qu'au moment de développer le Mondex original, nous ne nous sommes pas intéressés à l'aspect authentification, il n'y avait donc pas besoin d'avoir l'attribut t_bld_dst, il suffisait d'avoir l'attribut t_dst qui contenait le porte-monnaie destination. D'où le besoin de renommer dans une première étape l'attribut t_dst en t_bld_dst. En effet, un porte-monnaie n'est plus sûr de l'identité de la destination.

4.1 Renommage

Un renommage est également nécessaire pour l'ensemble porteur *Agent* du modèle du protocole cryptographique car après combinaison, les agents correspondent dans ce cas aux porte-monnaie. Dans ce cas, les types *Agent* et porte-monnaie sont des types de base qui ont tous les deux les mêmes propriétés. Il en est de même pour t_dst et t_bld_dst qui sont toutes deux fonctions de même type avec les mêmes propriétés. Ces renommages n'ont que peu d'incidence sur les preuves déjà faites des modèles correspondants. Les étapes par lesquelles passe une transaction sont modélisées dans les deux modèles par des sous-ensembles de

T, ces variables sont indépendantes l'une de l'autre ce qui nous évite de changer chaque modèle pour considérer les étapes de l'autre modèle, ceci aurait été notamment le cas si nous avions exprimé les différentes étapes des transactions par des fonctions. Dans le cas où ces variables ont des types différents, ce qui est possible vu que les modèles ont été développés séparément, il faut envisager un raffinement supplémentaire pour le modèle où les étapes d'une transaction sont modélisées par une fonction et la remplacer par des variables de type ensemble.

La figure 6 montre la démarche globale entreprise pour combiner les deux modèles :

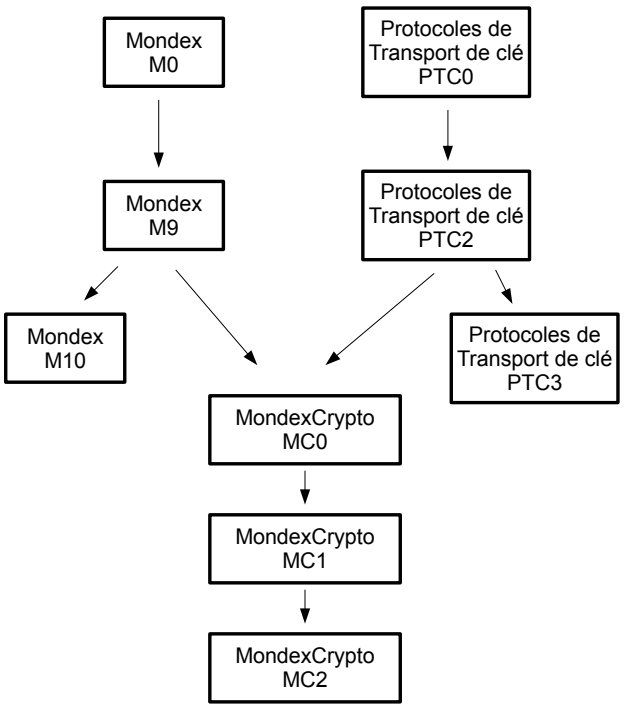

Fig. 6. La combinaison des deux modèles

Pour obtenir le modèle *MC0* du Mondex avec une couche cryptographique, nous avons combiné les modèles *M9* et *PTC2* et non pas *M10* et *PTC3* car les derniers raffinements de Mondex et du protocole cryptographique servent à supprimer les transactions pour arriver au modèle final. Ces deux raffinements sont de simples raffinements de données où les transactions sont remplacées par les numéros de séquences dans Mondex et par les nonces dans les protocoles cryptographiques. A chaque transaction abstraite est associée un numéro séquentiel unique contenu dans le porte-monnaie électronique dans le cas du Mondex

et un nonce dans le cas des protocoles cryptographiques. Les transactions abstraites sont nécessaires dans la démarche adoptée pour combiner les modèles, c'est pourquoi nous avons combiné les modèles *M9* et *PTC2* pour obtenir *MC0*. A l'issue de la combinaison un raffinement est effectué pour supprimer, finalement, les transactions abstraites.

4.2 Le modèle *MC0*

La combinaison a été obtenue par l'application d'un double raffinement. En effet, *MC0* raffine le modèle *M9* pour hériter des deux propriétés de sûreté de Mondex garantissant la non perte d'argent. Il raffine également le modèle *PTC2* garantissant les propriétés sur la clé de session donnée par le protocole de transport de clé et utilisée lors de la transaction. On retrouve dans le modèle résultant de ce double raffinement les événements de chacun des modèles combinés. Les événements du modèle du Mondex ont été renforcés avec la garde : $t\_dst(t) = t\_bld\_dst(t)$, comme dans l'événement *Event_DEB* présenté dans la section 2 :

EVENT Event_DEB
ANY
t
A
B
$money$
WHERE
$grd1 : t \in epr$
$grd2 : A = t\_src(t)$
$grd3 : B = t\_bld\_dst(t)$
$grd4 : money \in \mathbb{N}$
$grd5 : t\_dst(t) = t\_bld\_dst(t)$
...
THEN
$act1 : epr := epr \setminus \{t\}$
$act2 : epa := epa \cup \{t\}$
$act3 : conBalance(A) := conBalance(A) - money$
...
END

Nous voulons que dans le nouveau modèle résultant, le porte-monnaie destination soit authentifié à chaque étape. Pour ce faire les invariants correspondants ont été rajoutés. L'invariant suivant garantit, par exemple, pour le porte-monnaie source de créditer le porte-monnaie destination en étant sûr de son identité :

$$inv16 : \forall t \cdot t \in epa \Rightarrow t\_dst(t) = t\_bld\_dst(t)$$

4.3 Le modèle *MC1*

La garde $t\_dst(t) = t\_bld\_dst(t)$ du modèle *MC0* ne peut être laissée en l'état, le modèle *MC0* est raffiné pour remplacer cette garde. Grâce au modèle du protocole de transport de clé, chaque transaction se voit associer une clé de session sûre. Celle-ci est utilisée pour remplacer la garde précédente, un porte-monnaie ne teste plus directement s'il communique avec le bon porte-monnaie mais il teste si la clé avec laquelle le message reçu est crypté correspond à la clé associée à sa transaction courante. Les propriétés de la clé de session héritées du modèle *PTC2* permettent aisément de prouver l'obligation de preuve correspondant au fait que la garde concréte implique la garde abstraite.

Finalement les raffinements de données déjà appliqués séparément aux modèles du Mondex et des protocoles cryptographiques sont également appliqués au modèle *MC1* pour obtenir le modèle final *MC2* et supprimer les transactions abstraites.

L'outil utilisé pour l'ensemble du processus de modélisation est la plateforme RODIN [9]. Comme le double raffinement n'est pas pris en charge dans ce modèle, les deux raffinements et les preuves correspondantes ont été faits séparément pour chacun des deux modèles de Mondex et des protocoles cryptographiques. Le tableau suivant donne un bilan des preuves :

Table 1. Bilan des preuves pour le double raffinement.

Modèle	Nombre total de preuves	Automatiques	Interactives
Raffinement de Mondex	134	123 (91%)	11 (9%)
Raffinement cryptographique	63	52 (82%)	11 (18%)
Total	197	175 (88%)	22 (12%)

5 Conclusion

Ce travail illustre une technique de combinaison de deux modèles préalablement développés et prouvés, afin d'obtenir un modèle héritant des propriétés des deux systèmes combinés. La technique utilisée est celle du *double raffinement* basée sur la notion de *transactions abstraites*. Pour rendre le double raffinement possible, certaines modifications ont été apportées sur les deux modèles originaux pour les rendre compatibles notamment des renommages de variables, ainsi que des renommages d'ensembles porteurs. La technique a été utilisée pour combiner le système Mondex avec un protocole de transport de clé. Il faut noter que cette technique a été appliquée sur différents protocoles de transport de clé, car la technique de combinaison est indépendante du choix du protocole, d'autant plus que les détails concernant le protocole cryptographique utilisé dans la vraie carte Mondex déployée ne sont pas disponibles dans la littérature pour d'évidentes raisons d'ordre commercial. Grandy et ses coauteurs [7] ont appliqué aussi une

transformation sur leur développement initial du Mondex mais leur travail était beaucoup plus lié à une utilisation complète du prouveur pour démontrer les propriétés du Mondex crypté, sans réutiliser des preuves précédemment réalisées. Ainsi, l'effort de modélisation n'inclut pas l'effort de preuve, puisque la preuve est réalisée en fin de modélisation. Notre démarche vise à rationaliser l'effort de preuve et nous avons réutilisé la majorité des preuves des deux modèles combinés. Un élément qui n'est pas apparu dans notre travail est l'utilisation d'une autre technique pour obtenir le modèle du protocole cryptologique, puisque nous avons appliqué là aussi un schéma de combinaison du modèle d'attaque de Dolev-Yao et du protocole pour en faire un protocole de transport. Pour la suite, il reste à mettre en œuvre ces transformations et à poursuivre l'analyse de ces protocoles cryptologiques, en visant à en développer un réellement nouveau.

References

1. The Mondex electronic purse system. http://www.mondex.com.
2. Projet ANR-RIMEL. Proof-based design patterns. Livrable rimel, LORIA, Juillet 2008.
3. Nazim Benaïssa. Modelling attacker's knowledge for cascade cryptographic protocols. In Egon Börger, Michael Butler, Jonathan P. Bowen, and Paul Boca, editors, *ABZ2008*, volume 5238 of *Lecture Notes in Computer Science*. Springer, 2008.
4. Simon Blake-Wilson and Alfred Menezes. Entity authentication and authenticated key transport protocols employing asymmetric techniques. In *Proceedings of the 5th International Workshop on Security Protocols*, pages 137–158, London, UK, 1998. Springer-Verlag.
5. Michael Butler and Divakar Yadav. An incremental development of the mondex system in event-b. *Formal Asp. Comput.*, 20(1):61–77, 2008.
6. D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, Mar 1983.
7. Holger Grandy, Markus Bischof, Kurt Stenzel, Gerhard Schellhorn, and Wolfgang Reif. Verification of mondex electronic purses with kiv: From a security protocol to verified code. In Jorge Cuéllar, T. S. E. Maibaum, and Kaisa Sere, editors, *FM*, volume 5014 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2008.
8. T. Hoare. Grand Challenge GC6, 2005.
9. Christophe Metayer, Jean-Raymond Abrial, and Laurent Voisin. Event-B language. RODIN Project Deliverable D7, May 2005.
10. Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, 1978.
11. Lawrence C. Paulson. *Isabelle - A Generic Theorem Prover (with a contribution by T. Nipkow)*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.
12. Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.
13. Susan Stepney, David Cooper, and Jim Woodcock. An electronic purse: Specification, refinement, and proof. Technical monograph PRG-126, Oxford University Computing Laboratory, July 2000.