Réseaux 4G : anticiper la sécurité des systèmes de transactions sur mobile

Chrystel Gaber, Mohammed Achemlal, Baptiste Hemery, Marc Pasquet, Pascal Urien

Réseaux 4G : anticiper la sécurité des systèmes de transactions sur mobile

Chrystel Gaber*†
Mohammed Achemlal *†
Baptiste Hemery†
Marc Pasquet †
Pascal Urien‡

Résumé : Les services de transactions par mobile se sont beaucoup développés depuis quelques années. Parmi les plus célèbres, M-PESA connaît un succès indéniable puisque 70% des abonnés de téléphonie mobile du Kenya, où le service est déployé depuis 2007, en détiennent un compte en décembre 2011. Cependant, ces services sont limités au niveau de l'ergonomie et de la sécurité puisqu'ils utilisent l'USSD, Unstructured Supplementary Service Data et les SMS, Short Message Service pour transporter les instructions de paiement. De plus, le canal qui transporte ces deux types de messages est amené à disparaitre dans les réseaux cellulaires de quatrième génération, 4G. Il est possible d'encapsuler les USSD et les SMS dans des paquets IP mais cette solution n'exploite pas les fonctionnalités, tant au niveau de la sécurité et au niveau des usages, qu'offre l'évolution des technologies. Nous proposons une architecture pour les services de transactions sur mobiles qui assure une sécurité de bout-en-bout au niveau applicatif.

Mots Clés : Sécurité des transactions, paiement mobile, architecture sécurisée, réseaux 4G.

1 Introduction

L'écosystème des paiements mobiles s'est beaucoup développé ces dernières années et de nombreux modèles ont émergé. Parmi ceux-ci, les services de transfert d'argent par mobile ont pris de l'importance. Ainsi, M-PESA comptait environ 19 millions de souscrivants, soit environ 70% des abonnés mobiles au Kenya en décembre 2011 [CCK12]. Un autre exemple de service de transfert d'argent par mobile est Orange Money qui en 2012 était déployé dans dix pays et qui comprenait 14% des souscrivants mobiles dans ces pays [Ora12]. Cependant, ces services utilisent l'USSD, Unstructured Supplementary Service Data et les SMS, Short Message Service qui impliquent des limitations que nous montrons dans cet article. De plus, comme les canaux transportant ces services ne sont pas prévus dans les réseaux de quatrième génération, nous proposons d'adapter ces services et de tirer profit des fonctionnalités des nouvelles technologies pour résoudre des problèmes de sécurité. Notre objectif est de faire évoluer les protocoles existants pour garantir une sécurité de bout-en-bout entre l'application et la plateforme de transaction. Le but est

∗. France Télécom - Orange Labs, 38 rue des coutures, 14000 Caen, France {chrystel.gaber, mohamed.achemal}@orange.com

†. 1 Normandie Univ, France ; 2 UNICAEN, GREYC, F-14032 Caen, France; 3 ENSICAEN, GREYC, F-14032 Caen, France; 4 CNRS, UMR 6072, F-14032 Caen, France

‡. Telecom Paristech, UMR 5141, 37/39 rue Dareau 75014, Paris, France

aussi d'assurer l'anonymat du client vis-à-vis du marchand, ce qui n'est pas le cas dans les systèmes actuels. L'article est organisé comme suit. Tout d'abord, les services de transfert d'argent par mobile et les travaux antérieurs sont présentés. Ensuite, l'architecture proposée et les protocoles de paiement associés sont exposés. Finalement, l'article est conclu.

2 Les services de transfert d'argent par mobile

2.1 Architecture actuelle des services de transfert d'argent par mobile

Les services de transfert sur mobile comprennent plusieurs acteurs dont les relations sont illustrées FIGURE 1. La banque centrale et la banque commerciale permettent et contrôlent l'émission de monnaie électronique, m, par l'opérateur téléphonique qui possède le système. La monnaie électronique est ensuite distribuée à travers un réseau de grossistes à des détaillants ou à des fournisseurs de biens et de services. Ceux-ci la distribuent à leur tour aux porteurs qui réalisent des transferts ou des paiements.

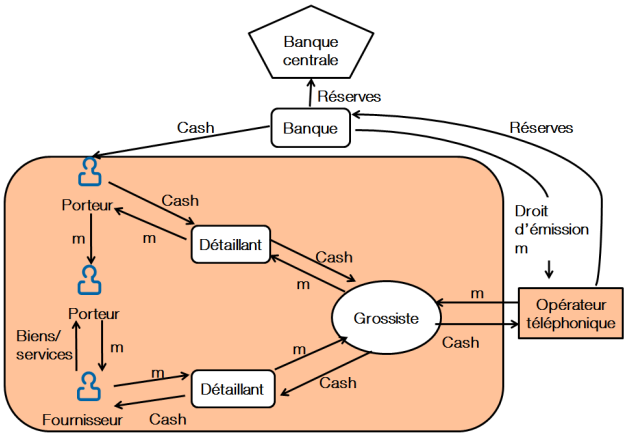


FIGURE 1: Relations entre les acteurs du paiement mobile prépayé

La plupart des acteurs représentés en figure 1, les porteurs, les fournisseurs de biens ou de services et certains détaillants, accèdent au service grâce à une application sur leur téléphone. Celle-ci communique avec le serveur distant par USSD et SMS en passant par le réseau opérateur. Certains acteurs peuvent accéder au service grâce à un ordinateur mais nous ne prenons pas en compte ce cas de figure.

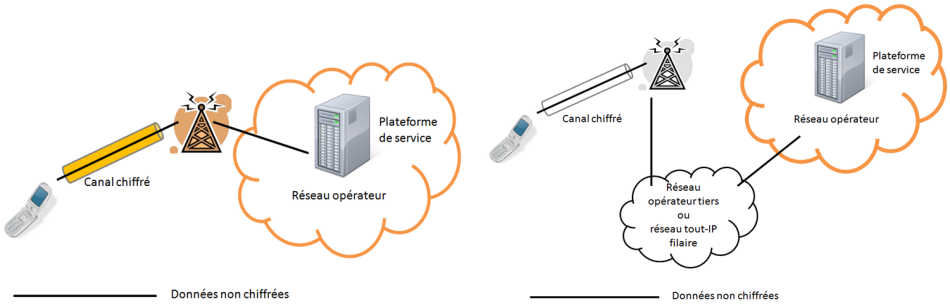
2.2 Limitations de l'architecture actuelle

L'USSD dispose d'un débit de 800 bits par seconde et implique que les menus de l'application de paiement sont envoyés de la plateforme vers le terminal mobile. La vitesse de transmission limite donc la taille des données pouvant être transmises et donc l'ergonomie. Cela pourrait être amélioré si les messages transitaient par l'ADSL qui a un débit de minimum de 1 Mbits par seconde ou la 4G avec un débit théorique maximal de 100 Mbits par seconde.

De plus, les réseaux 2G sont vulnérables aux attaques man-in the middle entre une antenne-relais GSM (BTS, Base Tranceiver Station) et un mobile car il n'y a pas d'authentification mutuelle

[NP09]. En outre, comme c'est l'antenne-relais qui choisit les protocoles de sécurité utilisés, elle peut imposer au mobile des protocoles de sécurité faible. Le matériel pour réaliser cette attaque est devenu abordable et facilement accessible [NP09]. Cependant, les terminaux utilisant la 3G ou la 4G sont encore vulnérables à cette attaque puisque, pour des raisons de compatibilité, ceux-ci peuvent se connecter à des antennes 2G.

Les données transmises par un mobile sont chiffrées uniquement entre le mobile et la première antenne rencontrée. Ensuite, ces données transitent en clair puisqu'elles sont censées être dans le réseau appartenant (et maîtrisé par) l'opérateur. Ce cas est illustré FIGURE 2)a). Il existe cependant des cas, comme l'itinérance, où l'opérateur fait appel à des réseaux qu'il ne maîtrise pas comme le montre FIGURE 2b). L'opérateur peut aussi faire passer ses données d'une antenne réseau vers un réseau filaire tout-IP qu'il ne maîtrise pas pour ensuite les faire revenir dans un réseau qu'il contrôle.



(a) Les données ne passent que par le réseau de l'opérateur à qui appartient la plateforme de service.

(b) Les données passent par un autre réseau que celui de l'opérateur à qui appartient la plateforme de service

FIGURE 2: Différentes manières de faire transiter des données vers une plateforme de service

Le canal transportant l'USSD n'est pas prévu dans les réseaux 4G. Il serait possible de continuer à créer des messages de type USSD et de les encapsuler dans des paquets IP cependant nous souhaitons proposer une méthode qui tire profit des capacités des réseaux tout-IP pour fournir une sécurité de bout-en-bout entre l'application de paiement dans l'élélement sécurisé et la plateforme de paiement de l'opérateur qui possède le service de transactions sur mobile.

L'anonymat du client vis-à-vis du marchand n'est donc pas assurée puisque, pour initier la transaction, le marchand saisit le numéro de téléphone du client.

3 Travaux antérieurs

Les architectures de paiement sur mobile se divisent en trois catégories selon l'interaction qui existe entre les différents acteurs du paiement [CKST01]. Si les trois acteurs sont connectés et interagissent lors de l'opération de paiement, l'architecture est tout-connecté. Si le payé et le payeur interagissent et un d'entre eux est connecté à la plateforme de paiement, l'architecture est semi-connecté, soit à travers le mobile du client soit à travers le mobile du marchand [CKST01]. Si le payeur et le payé interagissent mais aucun d'entre eux n'est connecté à la plateforme, l' architecture est déconnectée. Dans la suite, nous ne considérons que les deux premières architectures car nous proposons ici de modifier l'architecture existante en mode tout-connectée et d'y ajouter un mode semi-connecté.

La plupart des systèmes de transactions sur mobile déployés actuellement sont basés sur une architecture tout-connecté. Généralement, le marchand initie la transaction en envoyant au serveur le montant de la transaction et le numéro de téléphone de son client entre autres. Le client est ensuite contacté par la plateforme de paiement afin qu'il confirme le paiement. Nous considérons que le cas du transfert entre particuliers est un cas spécifique d'utilisation de cette architecture tout-conectée. Les porteurs et les marchands correspondent avec la plateforme de paiement à l'aide de SMS ou d'USSD. La sécurité de ces systèmes est basée sur les fonctionnalités du réseau mais comme nous l'avons vu précédemment, cela représente des limitations et des faiblesses.

Parmi les architectures tout-connecté pouvant être déployées sur des réseaux tout-IP, certaines, [GKR+09, HHH06, KHVC04, MY08, ZFM08], sont basées sur une infrastructure à clé publique, *PKI*. D'autres, [FBLR08, SS12], utilisent des clés secrètes pour sécuriser les communications. Seules les architectures présentées dans [HHH06, KHVC04] utilisent un élément sécurisé, *SE*. Ces deux architectures ne peuvent pas s'appliquer à notre contexte. Celle de [KHVC04] inclut les banques qui n'internviennent pas dans notre cas. Quant à celle de [HHH06], elle utilise les comptes bancaires des clients comme identifiant et ne garantit pas l'anonymat vis-à-vis du marchand.

Les systèmes de transactions sur mobile qui ont une architecture semi-connectée correspondent majoritairement aux cas où le mobile se substitue à la carte bancaire. Il s'agit par exemple d'une partie de l'application Google Wallet ou de Cityzi en France. Dans ce cas, un SE dans le mobile contient une application similaire à celle présente sur la carte bancaire. Ce SE dialogue ensuite avec un Terminal de Paiement Electronique, *TPE* chez le marchand. Le TPE dialogue avec la plateforme de paiement si nécessaire. La transaction suit alors le même parcours qu'une transaction réalisée par carte bancaire et la sécurité de la communication est assurée par les fonctionnalités du TPE. Cette approche n'est pas adaptée à notre contexte car nous prenons comme hypothèse que le marchand utilise lui aussi un terminal mobile et non un TPE. De plus, le système que nous considérons est un système 3-coins géré par un opérateur téléphonique. Toute l'infrastructure bancaire qui assure la sécurité des instructions de paiement après le TPE n'est donc pas la même dans notre cas. Les architectures proposées par [CHM+10, KLK09, NSCov] sont centrées sur le marchand et [IC07] propose une architecture soit centrée sur le porteur soit sur le marchand. Parmi elles, [CHM+10, IC07] ne spécifient pas où sont stockées les clés et les informations sensibles. Nguyen *et.al.* [NSCov] utilisent le mobile pour cela, que nous considérons non-sécurisé. Seule l'architecture proposée par [KLK09] se base sur l'utilisation d'un SE. Cette architecture n'est pas adaptée puisqu'elle se base sur les infrastructures bancaires existantes pour gérer la transaction. L'architecture proposée dans [CHM+10] n'est pas adaptée à notre cas d'usage puisqu'elle s'appuie sur des fonctionnalités des réseaux GSM et 3G dont nous cherchons à être indépendants. L'architecture de [IC07] n'est également pas adaptée puisqu'elle utilise des informations bancaires. A notre connaissance, aucun système de transactions sur mobile ne se base sur une architecture semi-connectée aujourd'hui.

Les systèmes de transactions sur mobile déployés aujourd'hui qui correspondent à notre contexte d'étude ont une architecture tout-connectée qui ne garantit pas l'anonymat du client. Ni les architectures de type tout-connecté ou semi-connecté existantes ou proposées dans la littérature ne répondent aux besoins exposés ici. Certaines d'entre elles ne permettent pas de garantir l'anonymat ou ne sont pas basées sur des SE pour garantir la sécurité. Il n'est donc pas possible de les utiliser pour l'adaptation des services de transaction sur mobile aux réseaux tout-IP.

4 Architecture proposée

D'après la partie précédente, il existe trois types d'architecture possibles selon les relations qui existent entre le payé, le payeur et la plateforme de paiement. Nous adaptons aux réseaux 4G l'architecture tout-connecté des services actuels pour les transferts entre particuliers et pour le paiement marchand en face à face. Cette adaptation implique que le client a un forfait lui permettant l'accès aux données ce qui peut limiter le développement du service. La deuxième contribution de cet article est donc l'extension des protocoles existants pour proposer un mode semi-connecté. Un mode tout-déconnecté limité permettant de réaliser des transactions même dans des zones où il n'y a pas de connexion réseau est aussi très intéressant mais n'est pas traité ici.

4.1 Hypothèses

Le marchand et le porteur disposent d'une paire de clés publique, privée ainsi que d'un certificat stockés dans leur carte SIM (Subscriber Identity Module). La plateforme de paiement dispose également d'une paire de clés publique, privée. Comme le MNO gère à la fois la plateforme, l'infrastructure de clés publiques et les cartes SIM, nous supposons que la clé publique de la plateforme et son certificat sont déployés dans les cartes SIM des porteurs et des marchands. Les certicats sont signés par la plateforme et peuvent donc être vérifiés par toute entité possédant la clé publique de la plateforme.

Nous ne faisons pas confiance aux systèmes d'exploitation déployés de base dans les téléphones. Pour établir la sécurité dans les téléphones, nous proposons d'utiliser conjointement un SE et un environnement d'exécution sécurisé, *TEE*. En effet, le SE offre un niveau de sécurité plus important que le TEE [Glo]. Il permet donc de gérer les secrets et les opérations les plus critiques de notre architecture. Le TEE permet un accès sécurisé aux périphériques du téléphone, clavier, écran, mémoire et autres. Il permet également d'avoir un environnement moins restreint et plus performant que les SE. Nous souhaitons utiliser la carte SIM en tant que SE. C'est le choix le plus intéressant et le plus pratique étant donné que cet élément est entièrement sous la responsabilité de l'opérateur qui gère le système de transfert d'argent sur mobile.

4.2 Protocoles

La TABLE 1 regroupe les notations utilisées dans cette partie. Certains des protocoles ci-dessous supposent qu'un canal sécurisé est établi et qu'une clé secrète de session est partagée entre deux entités. Nous utilisons pour cela le protocole IKEv2 [IETa] constitué de deux phases. Dans la première a lieu un échange Diffie Hellmann pour calculer une clé de session. La seconde phase est composée de deux échanges où les deux parties s'authentifient mutuellement en envoyant leur certificat respectif, les deux messages échangés précédemment, un aléa, généré par l'autre partie et le haché de ces 2 derniers éléments signé avec leur clé privée respective. La vérification de cette signature permet d'authentifier chacune des deux parties. Les messages échangés dans cette deuxième phase sont chiffrés avec la clé secrète partagée. Celle-ci est ensuite utilisée pour sécuriser les échanges des protocoles de paiement exposés ici. Ceux-ci suivent le format ESP et comprennent une entête HDR, tous deux définis dans [IETd, IETa]. La clé de session peut être rafraichie de la manière décrite dans le protocole IKEv2. Une politique de sécurité définit les modalités de ce rafraichissement.

4.2.1 Transfert entre particuliers

Le prototocole proposé pour le transfert d'argent entre particuliers est représenté FIGURE 3. Tout d'abord, l'envoyeur initie la demande de paiement en saisissant le montant et l'identité du

M, P, E, D	marchand, porteur, expéditaire ou destinataire	Pp	plateforme de paiement
PK_x	clé publique de x	pk_x	clé privée de x
$\{m\}_{PK_x}$	m chiffré PK_x	ID_x	Identité de x
Not	Notification	$alea$	Nombre aléatoire
TA	Montant de la transaction	.	opérateur de concaténation
$a.b.c.SIG_x$	Signature par x du haché de a.b.c	$SIG_x(m)$	Sigature par x du haché de m
$PreBal_x$	Solde de x avant la transaction	$PostBal_x$	Solde de x après la transaction
$hash()$	Fonction de hachage	TT	Type de transaction
HDR	Entête des messages au format ESP	SK_{xy}	Clé secrète partagée par x et y
PMK	Clé pré-maître	$CERT(x)$	Certificat de x
$Verif_x$	Valeur de la vérification du code secret de x	TID	Numéro d'identification de la transaction
Req_z/Rep_z	Requête / réponse de z=Val, Tran, ID, Proposition, Choix et Défi	Val	Validation
$Tran$	Transaction	ID	Identité

TABLE 1: Symboles utilisés

destinataire. Sa carte SIM établit alors un canal sécurisé avec la plateforme de paiement et calcule la clé de session SK_{PPp}. La carte SIM envoie ensuite la requête de transaction contenant les détails de la transaction. A la réception de ce message, la plateforme crée un identifiant de transaction et demande à la SIM du client de valider cette transaction. Cette requête est accompagnée d'un récapitulatif et de l'état du compte du porteur. La carte SIM fait afficher par le mobile ces éléments au client afin d'obtenir sa validation. Ici, la validation correspond à l'entrée d'un code secret spécifique à l'application. Ce code est vérifié par la carte SIM qui envoie ensuite une preuve de cette validation à la plateforme. Celle-ci est constituée de la valeur $Verif_P$ qui indique l'état de la validation et de $SIG_P(Verif_P, alea_2)$ qui permet de vérifier l'intégrité, l'authenticité et le non-rejeu de la preuve de validation du porteur. Finalement, une notification est envoyée au porteur pour confirmer le paiement.

4.2.2 Transaction en face à face en mode connecté

Le prototocole proposé pour réaliser une transaction en face à face en mode connecté est représenté FIGURE 4. Ce type de transaction correspond à un paiement marchand, un retrait ou un dépôt d'argent. Le marchand initie la transaction en saisissant un montant. Sa carte SIM établit alors un canal sécurisé avec la plateforme de paiement et une clé de session partagée SK_{PpM}. Elle envoie alors une requête de transaction à la plateforme en indiquant l'identité du marchand, le montant et identifiant de la transaction temporaire propre au marchand. La plateforme de paiement crée alors une transaction dans sa base de données et un numéro d'identification de la transaction. La plateforme répond avec une requête d'identification du client qui rappelle l'identifiant de transaction temporaire, un cookie, et la signature par la plateforme de celui-ci. Le marchand s'assure que le cookie provient bien de la plateforme et le transmet avec sa signature au porteur. Cette transmission se fait par un moyen de communication de proximité tel que le NFC (Near Field Communication) ou un canal de communication hors-bande tel qu'un QRcode. Le client s'assure à son tour que le cookie provient bien de la plateforme de paiement à l'aide de la signature et établit à son tour un canal sécurisé avec la plateforme. La carte SIM du porteur renvoie alors à la plateforme le cookie, son identité et la signature de son identité. La plateforme de paiement s'assure ensuite que le cookie correspond bien à une transaction en cours, que la date de validité du cookie n'est pas dépassée, et que l'identité du client correspond bien à la signature produite. La

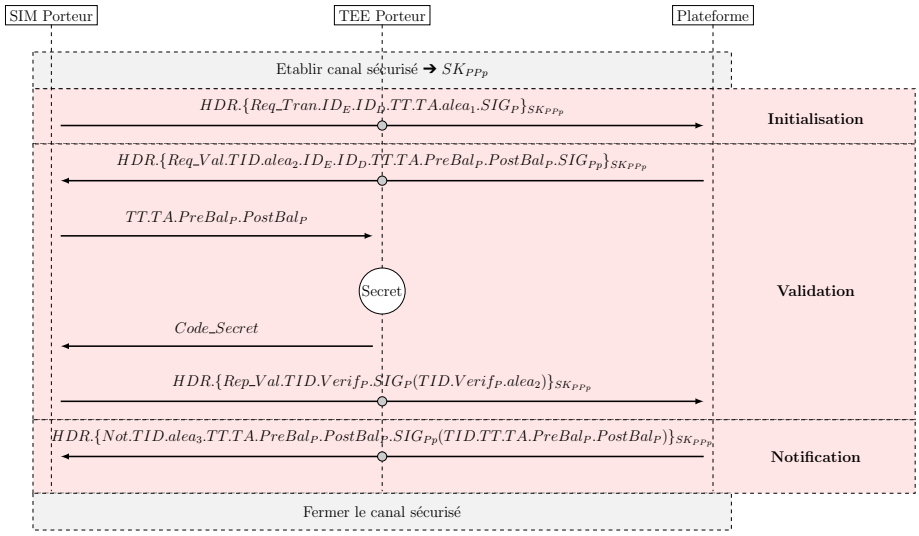


FIGURE 3: Protocole pour le transfert C2C

plateforme complète ensuite la transaction créée avec l'identité du client et envoie une demande de validation au marchand et au client. La validation et la notification se déroulent pour le porteur et le marchand de la même manière que dans le protocole de transfert d'argent entre particuliers.

Le cookie identifie de manière unique la transaction. Il comprend également une date, ce qui permet à la plateforme de rejeter des cookies trop anciens. Il est composé de deux parties. La première est compréhensible uniquement par la plateforme et la seconde permet d'authentifier le cookie sans divulguer les détails de la transaction. Nous proposons par exemple comme valeur de cookie : $\{TID.IDm.TA.t_{valid}.alea\}_{Pk_{Pp}}.\{h(TID.IDm.TA.t_{valid}.alea_{Pk_{Pp}})\}_{pk_{Pp}}$, où t_{valid} correspond à une date de validité du cookie.

4.2.3 Transaction en face à face en mode semi-connecté

L'objectif est de faire bénéficier au porteur l'accès aux données du marchand afin qu'il se retrouve dans le mode semi-connecté. Pour cela, nous adaptons le protocole EAP-TTLS, Extensible Authentication Protocol Tunneled Transport Layer Security [IETe]. Cette adpatation est représentée en FIGURE 5 pour le cas où la demande de connexion réussit.

Nous utilisons le terminal du marchand en tant que point d'accès et la plateforme de paiement en tant que serveur AAA (Authentication Authorization Accounting) et serveur TTLS (Tunneled Transport Layer Security). Pour plus de facilité ici, nous considérons le protocole Radius mais les principes présentés ici sont adaptables à tout protocole AAA. Nous supposons que c'est le TEE dans le terminal marchand qui prend en charge les différents messages présentés ici. En particulier, nous supposons que le niveau de sécurité du TEE suffit pour stocker les clés liées au protocole AAA et pour gérer l'accès des porteurs au réseau. Ses capacités sont plus importantes que la SIM et cela permet de ne pas trop alourdir le protocole. De plus, le TEE gère directement et de manière sécurisée les périphériques dont ceux permettant l'accès au réseau.

Les messages entre le terminal marchand et la SIM ou la plateforme de paiement et la SIM

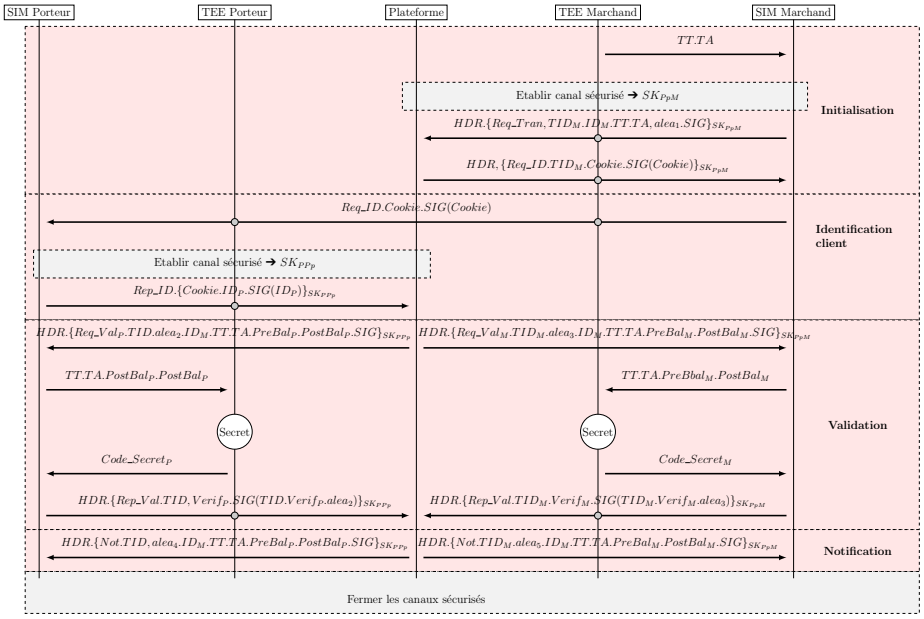


FIGURE 4: Protocole pour le paiement de proximité en mode connecté

sont des requêtes et réponses EAP [IETc]. Les différents messages EAP sont encapsulés dans le protocole ISO 7816 [ISO] entre la SIM client et le terminal client, dans un protocole LAN entre les terminaux client et marchand et dans le protocole RADIUS entre le terminal marchand et la plateforme. Ce sont les terminaux clients et marchands qui prennent les différentes traductions en charge. Conformément au protocole RADIUS [IETb], le terminal marchand et la plateforme de paiement ont une clé prépartagée S_{AN}. Celle-ci est utilisée en particulier pour chiffrer le message qui informe le terminal marchand du refus ou de l'acceptation par la plateforme de partager la connexion.

Ce protocole se découpe en quatre phases. Tout d'abord, la demande de connexion est initialisée et la SIM porteur transmet l'identité du porteur à la plateforme. Afin de préserver l'anonymat du porteur vis-àvis du marchand, nous transmettons l'identité du porteur, un aléa, la signature de ces deux éléments par la SIM porteur chiffrés avec la clé publique de la plateforme de paiement. Ensuite, un tunnel TLS est mis en place. Les messages M1 et M2 correspondent à la poroposition et au choix des fonctionnalités cryptographiques. La SIM porteur génère la clé prémaître PMK et calcule la clé secrète K à partir de PMK et des aléas $alea_{Pp}$, $alea_P$ échangés dans M1 et M2. La clé pré-maître chiffrée par la clé secrète du marchand est envoyée ainsi que le haché de M1 et M2 chiffré par K. La plateforme doit alors retrouver PMK et recalculer K pour initier la phase d'authentification du porteur. La plateforme et la SIM porteur vérifient chacun que les messages M1 et M2 reçus sont intègres. Ces différents échanges permettent d'authentifier la plateforme auprès du client et de garantir l'intégrité des messages M1 et M2. La clé K est utilisée pour sécuriser les prochains messages. La réponse d'identité du porteur est renvoyée au serveur qui transmet un défi à la SIM porteur. Celle-ci s'authentifie auprès du serveur en transmettant son certificat et en signant le défi. Si cette authentification réussit, la plateforme autorise le terminal marchand à par-

tager sa connexion avec le terminal client. Cet échange suit le format défini dans [IETe]. Il s'agit d'un message chiffré avec la clé secrète S_{AN} de type EAP-Success, encapsulé dans un message RADIUS Access-Accept en cas de succès, ou Access-Reject sinon.

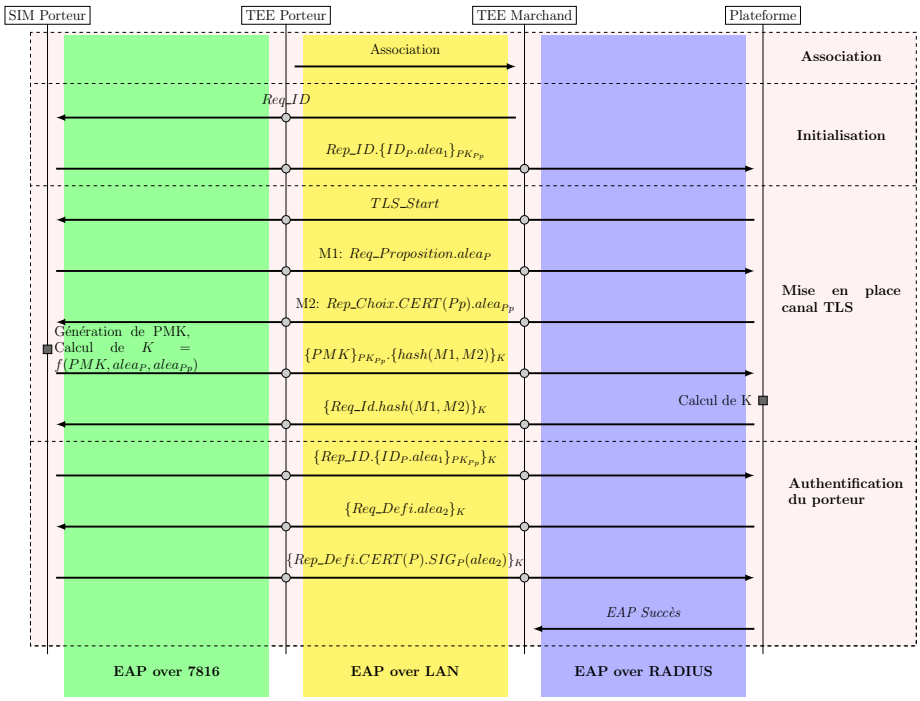


FIGURE 5: Protocole pour le mode semi-connecté

5 Conclusion et perspectives

Nous proposons une architecture qui s'adapte aux réseaux cellulaires de quatrième génération. Elle garantit une sécurité de bout-en-bout entre l'application et la plateforme de paiement. De plus, elle introduit l'anonymat du client envers le marchand lors d'un paiement. Les protocole de transfert entre particuliers et de paiement en face à face ont été adaptés. Un mode supplémentaire a été proposé. Il permet aux clients de profiter du service même s'ils ne disposent pas d'un forfait permettant l'accès aux données. Dans la suite, nous proposerons une extension du service pour permettre le paiement hors ligne. Tous les protocoles seront vérifiés formellement et implémentés.

Références

[CCK12] CCK. Quarterly sector statistics report. Technical report, Communications Commission of Kenya, 2012.

[CHM+10] W. Chen, G. P. Hancke, K. E. Mayes, Y. Lien, and J.-H. Chiu. Nfc mobile transactions and authentication based on gsm network. In *2010 Second International Workshop on Near Field Communication*.

[CKST01] Suresh Chari, Parviz Kermani, Sean Smith, and Ros Tassiulas. Security issues in m-commerce : A usage-based taxonomy. e-commerce agents. 2001.

[FBLR08] Tan Soo Fun, Leau Yu Beng, J. Likoh, and R. Roslan. A lightweight and private mobile payment protocol by using mobile network operator. In *Computer and Communication Engineering*, 2008.

[GKR+09] J. Gao, V. Kulkarni, H. Ranavat, L. Chang, and H. Mei. A 2d barcode-based mobile payment system. In *Multimedia and Ubiquitous Engineering*, 2009.

[Glo] Global Platform. Globalplatform made simple guide : Trusted execution environment (tee) guide. http://www.globalplatform.org/mediaguidetee.asp. Last visited on 17/04/2013.

[HHH06] Marko Hassinen, Konstantin Hyppoenen, and Keijo Haataja. An open, pki-based mobile payment system. In *Emerging Trends in Information and Communication Security*, 2006.

[IC07] Jesus Tellez Isaac and Jose Sierra Camara. A secure payment protocol for restricted connectivity scenarios in m-commerce. In *8th international conference on E-commerce and web technologies*, 2007.

[IETa] IETF Internet Engineering Task Force. Rfc 5996 internet key exchange protocol version 2 (ikev2).

[IETb] IETF Network Working Group. Rfc 2865 remote authentication dial in user service (radius).

[IETc] IETF Network Working Group. Rfc 3748 extensible authentication protocol (eap).

[IETd] IETF Network Workting Group. Rfc 4301 security architecture for the internet protocol.

[IETe] IETF Network Workting Group. Rfc 5281 extensible authentication protocol tunneled transport layer security authenticated protocol version 0.

[ISO] ISO. Iso/iec 7816-4 :2013 identification cards – integrated circuit cards – part 4 : Organization, security and commands for interchange.

[KHVC04] S. Karnouskos, A. Hondroudaki, A. Vilmos, and B. Csik. Security, trust and privacy in the secure mobile payment service. In *3rd International Conference on Mobile Business*, 2004.

[KLK09] Kiran S. Kadambi, Jun Li, and Alan H. Karp. Near-field communication-based secure mobile payment service. In *11th International Conference on Electronic Commerce*.

[MY08] J. Meng and L. Ye. Secure mobile payment model based on wap. In *Wireless Communications, Networking and Mobile Computing*, 2008.

[NP09] Karsten Nohl and Chris Paget. Gsm - srsly ? Presented at 26C3 in Berlin, http://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf, December 2009.

[NSCov] T. N T Nguyen, P. Shum, and E. H. Chua. Secure end-to-end mobile payment system. In *Mobile Technology, Applications and Systems*, 2005.

[Ora12] Orange. Orange money dépasse les 4 millions de clients et lance ses services en jordanie et à l'ile maurice. http://www.orange.com/fr/presse/communiques/communiques-2012/Orange-Money-depasse-les-4-millions-de-clients-et-lance-ses-services-en-Jordanie-et-a-l-Ile-Maurice, juin 2012. Last visited on 12/04/2013.

[SS12] V.C. Sekhar and M. Sarvabhatla. Secure lightweight mobile payment protocol using symmetric key techniques. In *Computer Communication and Informatics*, 2012.

[ZFM08] Ge Zhang, Cheng Feng, and Christoph Meinel. Simpa : A sip-based mobile payment architecture. In *Seventh IEEE/ACIS International Conference on Computer and Information Science*, 2008.