Paiement par carte bancaire (vue externe)

Acteurs (ou agents) :

- (A)lice
- (C)arte bancaire (possédée par A)
- (T)erminal du commerçant
- (B)anque (banque émettrice de la carte)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T
- T authentifie C ''Action Authentification''
- A donne son code (3456) à C
  Si m dépasse 1000 dinars
- T demande l'autorisation à B pour C
- B donne l'autorisation

Paiement par carte bancaire (vue interne)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : PKb
  - une clé privée :SKb
  - une clé symétrique partagée avec C : Kcb
- (C)arte possède des informations **publiques**
  - Data= nom, prénom, numéro carte, date de validité
  - Valeur de Signature VS = {hash(Data)} SKb
  - et la clé secrète Kcb
- (T)erminal possède
  - Fonction hash
  - une clé publique PKb

Phase hors ligne de la transaction :

Dans l'étape initiale, le terminal procède à l'authentification de la carte en calculant le hachage des données (DATA). Ensuite, il déchiffre la valeur de la signature VS en utilisant la clé publique de la banque. Si ces deux valeurs sont égales, alors il passe à l'étape suivante.

1- T authentifie C :1- Data , {hash(Data)}SKb

2- hash(Data) = hash(Data) du VS

1-Le terminal demande le code PIN à l'utilisateur : Le terminal (T) affiche un message demandant à l'utilisateur (A) d'entrer son code PIN. Ce code est une mesure de sécurité supplémentaire pour vérifier que la personne utilisant la carte est bien le détenteur légitime de la carte.

2-L'utilisateur entre son code PIN : L'utilisateur (A) saisit son code PIN (dans cet exemple, 3456) sur le clavier du terminal. Ce code est transmis au terminal (T).

3-Le terminal envoie le code PIN à la carte : Le terminal (T) envoie ensuite le code PIN à la carte (C). La carte a la capacité de vérifier si le code PIN est correct.

4-La carte confirme que le code PIN est correct : Si le code PIN entré par l'utilisateur est correct, la carte (C) envoie un message de confirmation ("ok") au terminal (T). Cela indique au terminal que le code PIN était correct, et la transaction peut continuer.

2- C authentifie A (A donne son code à C)
   - M1 :T → A : code ?
   - M2 :A→ T : 3456
   - M3 :T :→ C : 3456
   - M4 :C → T : ok

Si le montant est supérieur à 100 dinars T demande l'autorisation à B pour C

-La banque envoie un nombre aléatoire au terminal (T) : La banque (B) génère un nombre aléatoire (Nb) et l'envoie au terminal. Ce nombre aléatoire est utilisé pour garantir la sécurité des transactions.

-Le terminal transmet le nombre aléatoire à la carte (C) : Le terminal (T) reçoit le nombre aléatoire (Nb) de la banque et le transmet à la carte. Cela permet à la carte de connaître le nombre aléatoire qui a été généré.

-La carte envoie l'identité de A et le nombre aléatoire crypté au terminal (T) : La carte (C) crypte le nombre aléatoire (Nb) à l'aide d'une clé partagée (Kcb) qu'elle partage avec la banque (B). La carte envoie ensuite cette valeur cryptée, ainsi que l'identité d'Alice (A) au terminal.

-Le terminal transmet les informations à la banque (B) : Le terminal (T) envoie à la banque (B) les informations reçues de la carte, c'est-à-dire l'identité d'Alice (A) et le nombre aléatoire crypté ({Nb}Kcb). La banque peut alors déchiffrer le nombre aléatoire avec la clé partagée et vérifier les informations, et après la banque donne l'autorisation

3- B réalise l'authentification en ligne de C
   - T → B : Demande d'authentification
   - B→ C : Nb
   - C→ T : A, {Nb}Kcb
   - T→ B : A, {Nb} Kcb
   - B donne l'autorisation

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur : la non réplicabilité de la carte ,le secret autour des clés employées, du protocole, etc….

Faiblesse logique du protocole : Dans le cas de cette vulnérabilité du protocole, si un utilisateur crée une copie de sa carte bancaire et modifie certains paramètres de cette copie, la carte pourrait accepter n'importe quel numéro PIN qui est saisi.

4- C → T : Data, {hash(Data)}SKb
5- $T \rightarrow A$: code ?
6- A → C : 7575
7- C → T : ok

Correction appliques par EMVCO (Europay, Mastercard, and Visa) en 2004

Europay, MasterCard et Visa ont produit SDA(Static Data Authentication), DDA(Dynamic Data Authentication), CDA(Combined Dynamic Data Authentication) pour les cartes bancaires.

1- SDA=Static Data Authentication

Dans SDA Le certificat {PKb }SKs contient la clé publique de la banque, qui a été cryptée avec la clé privée de l'autorité de certification , qui, comme nous l'avons vu dans les cours précédents, est aussi appelée le serveur d'authentification S.

- (A) lice possède un code secret : 3456
- (C)arte possède des informations publiques
  - Data= nom, prénom, numéro carte, date de validité
  - Valeur de Signature VS = {hash(Data)} SKb
  - Certificat {PKb }SKs de la clé de la banque
- (T)erminal possède
  - Fonction hash
  - une clé publique PKs

Phase hors ligne de la transaction :

Durant la première étape, le terminal décode le certificat pour obtenir la clé publique de la banque. Par la suite, le terminal authentifie la carte en calculant le hachage des données, DATA. Il décode ensuite la valeur de la signature, notée VS, en utilisant la clé publique de la banque. Si les deux valeurs obtenues correspondent, le terminal peut alors passer à l'étape suivante C authentifie A.

- T authentifie C : 1- {PKb }SKs , Data , {hash(Data)}SKb

  2- PKb , hash(Data) = hash(Data) du VS

Et Après , C authentifie A

-- C authentifie A (A donne son code à C)

- M1 :T → A : code ?
- M2 :A→ T : 3456
- M3 :T :→ C : 3456
- M4 :C → T : ok

Deuxième amélioration DDA CDA

- (A) lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - clé privee SKc propre a la (C)arte
- (C)arte possède des informations **publique**
  - Data= nom, prénom, numéro carte, date de validité
  - Valeur de Signature VS = {hash(Data)} SKb
  - Certificat {PKb }SKs de la clé de la banque
  - Certificat {PKc }SKb de la clé de la Carte
- (T)erminal possède
  - Fonction hash
  - une clé publique PKs

Phase hors ligne de la transaction :

Durant la première étape, le terminal décode le certificat de la banque pour obtenir la clé publique de la banque. Par la suite, décode le certificat de la Carte par la clé publique de Banque pour obtenir la clé publique de la Carte. Après le terminal authentifie la carte en calculant le hachage des données, DATA. Il décode ensuite la valeur de la signature, notée VS, en utilisant la clé publique de la banque. Si les deux valeurs obtenues correspondent, le terminal peut alors passer à l'étape suivante.

T authentifie C : 1-{PKb }SKs, {PKc }SKb , Data , {hash(Data)}SKb

2- PKb , PKc , hash(Data) = hash(Data) du VS

Une fois la phase d'authentification terminée, le terminal (T) génère un numéro aléatoire (Nt) qu'il envoie à la carte (C). La carte crypte ce numéro en utilisant sa clé privée et le renvoie au terminal. Le terminal, à son tour, décrypte ce message à l'aide de la clé publique de la carte qu'il a obtenue du certificat de la carte lors de l'étape précédente. Il vérifie ensuite si la valeur obtenue correspond à la valeur qu'il a lui-même générée. Si les deux valeurs sont égales, le terminal peut passer à l'étape suivante C authentifie A.

- T → C : Nt
- C → T : {Nt}SKc

C authentifie A

Pendant cette étape, le terminal demande à l'utilisateur A de saisir son code PIN secret. Une fois ce code entré par l'utilisateur, le terminal le crypte à l'aide de la clé publique de la carte puis le renvoie à celle-ci. La carte, de son côté, décrypte le message en utilisant sa clé privée et vérifie si le code reçu correspond au code PIN de l'utilisateur. Si les codes sont identiques, la carte envoie un message 'ok' pour confirmer l'authentification de l'utilisateur A.

1- $T \rightarrow A$: code ?
2- A → T : 3456
3- T → C : {3456}PKc
4- C→ T: OK