Smart Carduri

Prof. Dr. Adrian Atanasiu

Universitatea București

March 2, 2017

1 Prezentare generală

2 Hardware-ul unui smartcard

3 Funcțiile unui smartcard

4 Sistemul de operare pentru smartcard
- Structurile de date ale protocolului de aplicație (*APDU*)

5 Entități implicate în construcția/utilizarea unui smartcard

6 Exemple de partajare a securității sistemelor de carduri

7 Componente criptografice

8 Aspecte legate de securitate

9 Criptanaliza smartcardurilor
- Tehnici invazive de atac
- Tehnici non-invazive de atac

Un smartcard este un sistem de calcul portabil (sau detașabil) utilizat în efectuarea unor tranzacții, cu competențe sporite pentru a prezenta sau autentifica o identitate.

Un smartcard este un sistem de calcul portabil (sau detașabil) utilizat în efectuarea unor tranzacții, cu competențe sporite pentru a prezenta sau autentifica o identitate.

Smartcardurile au fost inventate și patentate în anii 70. Roland Moreno a patentat conceptul de "*card cu memorie*" în 1974. În 1977, Michel Ugon (Honeywell Bull) a inventat primul microprocesor de smartcard. Tot Honeywell Bull patentează în 1978 *SPOM* (*Self Programmable One-chip Microcomputer*) care definește arhitectura unui cip de smartcard.

Un smartcard este un sistem de calcul portabil (sau detașabil) utilizat în efectuarea unor tranzacții, cu competențe sporite pentru a prezenta sau autentifica o identitate.

Smartcardurile au fost inventate și patentate în anii 70.
Roland Moreno a patentat conceptul de "*card cu memorie*" în 1974. În 1977, Michel Ugon (Honeywell Bull) a inventat primul microprocesor de smartcard. Tot Honeywell Bull patentează în 1978 *SPOM* (*Self Programmable One-chip Microcomputer*) care definește arhitectura unui cip de smartcard.
Prima utilizare pe scară largă a cardurilor a fost *Télécarte* – cartelă telefonică prepaid folosită în Franța începând cu 1983.

Un smartcard este un sistem de calcul portabil (sau detașabil) utilizat în efectuarea unor tranzacții, cu competențe sporite pentru a prezenta sau autentifica o identitate.

Smartcardurile au fost inventate și patentate în anii 70.
Roland Moreno a patentat conceptul de "*card cu memorie*" în 1974. În 1977, Michel Ugon (Honeywell Bull) a inventat primul microprocesor de smartcard. Tot Honeywell Bull patentează în 1978 *SPOM* (*Self Programmable One-chip Microcomputer*) care definește arhitectura unui cip de smartcard.
Prima utilizare pe scară largă a cardurilor a fost *Télécarte* – cartelă telefonică prepaid folosită în Franța începând cu 1983.

Utilizarea cardurilor explodează în anii 90, odată cu introducerea în telefonia mobilă *GSM* a smartcardurilor bazate pe *SIM*.

Clasificare

- *Carduri (cartele) magnetice:*
  Conțin o unitate de memorie relativ mică (maxim 100 octeți) sub forma unei benzi magnetice și sunt destinate unei singure aplicații. Exemple: cartele telefonice, carduri hoteliere.

Clasificare

- *Carduri (cartele) magnetice:*
  Conțin o unitate de memorie relativ mică (maxim 100 octeți) sub forma unei benzi magnetice și sunt destinate unei singure aplicații. Exemple: cartele telefonice, carduri hoteliere.
- *Carduri cu procesor (smartcarduri)*:
  Conțin un mini-procesor, ceeea ce le dă posibitatea de a rula mai multe aplicații prezente pe card.

Clasificare

- *Carduri (cartele) magnetice:*
  Conțin o unitate de memorie relativ mică (maxim 100 octeți) sub forma unei benzi magnetice și sunt destinate unei singure aplicații. Exemple: cartele telefonice, carduri hoteliere.
- *Carduri cu procesor (smartcarduri)*:
  Conțin un mini-procesor, ceeea ce le dă posibitatea de a rula mai multe aplicații prezente pe card.
- *Carduri hibride*:
  O combinație între celelalte două tipuri.

Clasificare

- *Carduri (cartele) magnetice:*
  Conțin o unitate de memorie relativ mică (maxim 100 octeți) sub forma unei benzi magnetice și sunt destinate unei singure aplicații. Exemple: cartele telefonice, carduri hoteliere.
- *Carduri cu procesor (smartcarduri)*:
  Conțin un mini-procesor, ceeea ce le dă posibitatea de a rula mai multe aplicații prezente pe card.
- *Carduri hibride*:
  O combinație între celelalte două tipuri.
  În prezent există carduri hibride care conțin simultan micro-cip, bandă magnetică, cod de bare, memorie optică, poză și tabelă de semnătură.

Clasificare

Din punct de vedere al modului de legătură cu exteriorul:

Clasificare

Din punct de vedere al modului de legătură cu exteriorul:

- *Carduri cu contacte*:
  Transferul de date se face prin intermediul unui cititor de card, cu care se stabilește un contact direct.

Clasificare

Din punct de vedere al modului de legătură cu exteriorul:

- *Carduri cu contacte*:
  Transferul de date se face prin intermediul unui cititor de card, cu care se stabilește un contact direct.
- *Cardurile fără contacte*:
  Conțin – pe lângă un cip cu micro-procesor – un dispozitiv de emisie-recepție radio cu antenă.

Clasificare

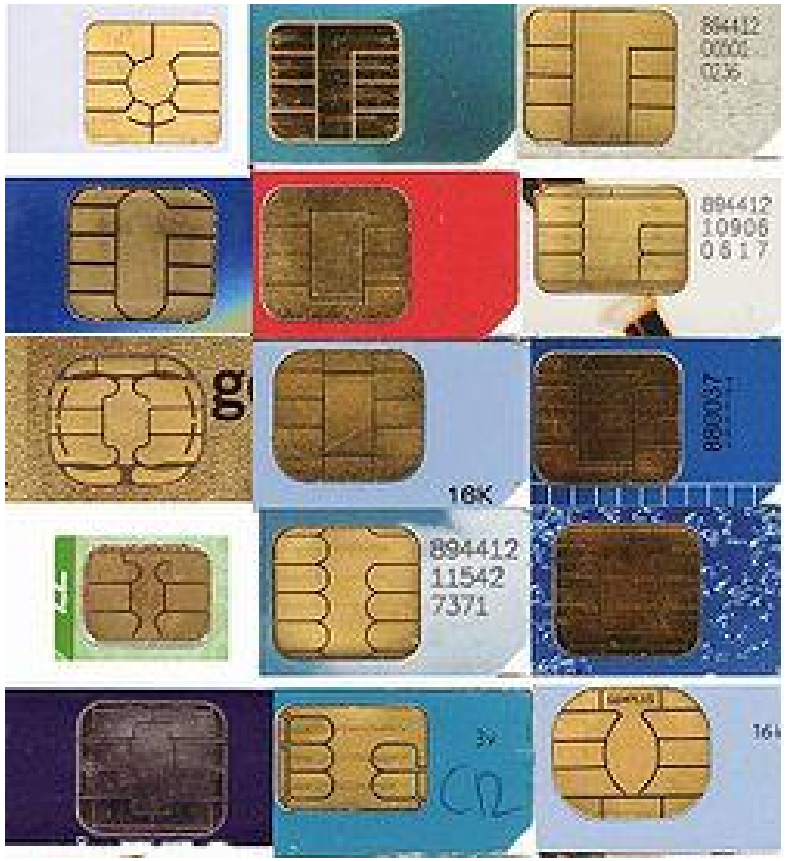
Din punct de vedere al modului de legătură cu exteriorul:

- *Carduri cu contacte*:
  Transferul de date se face prin intermediul unui cititor de card, cu care se stabilește un contact direct.
- *Cardurile fără contacte*:
  Conțin – pe lângă un cip cu micro-procesor – un dispozitiv de emisie-recepție radio cu antenă.
  Aceste carduri sunt de obicei dedicate unor aplicații specifice cum ar fi pentru transporturi sau pentru pașapoarte.

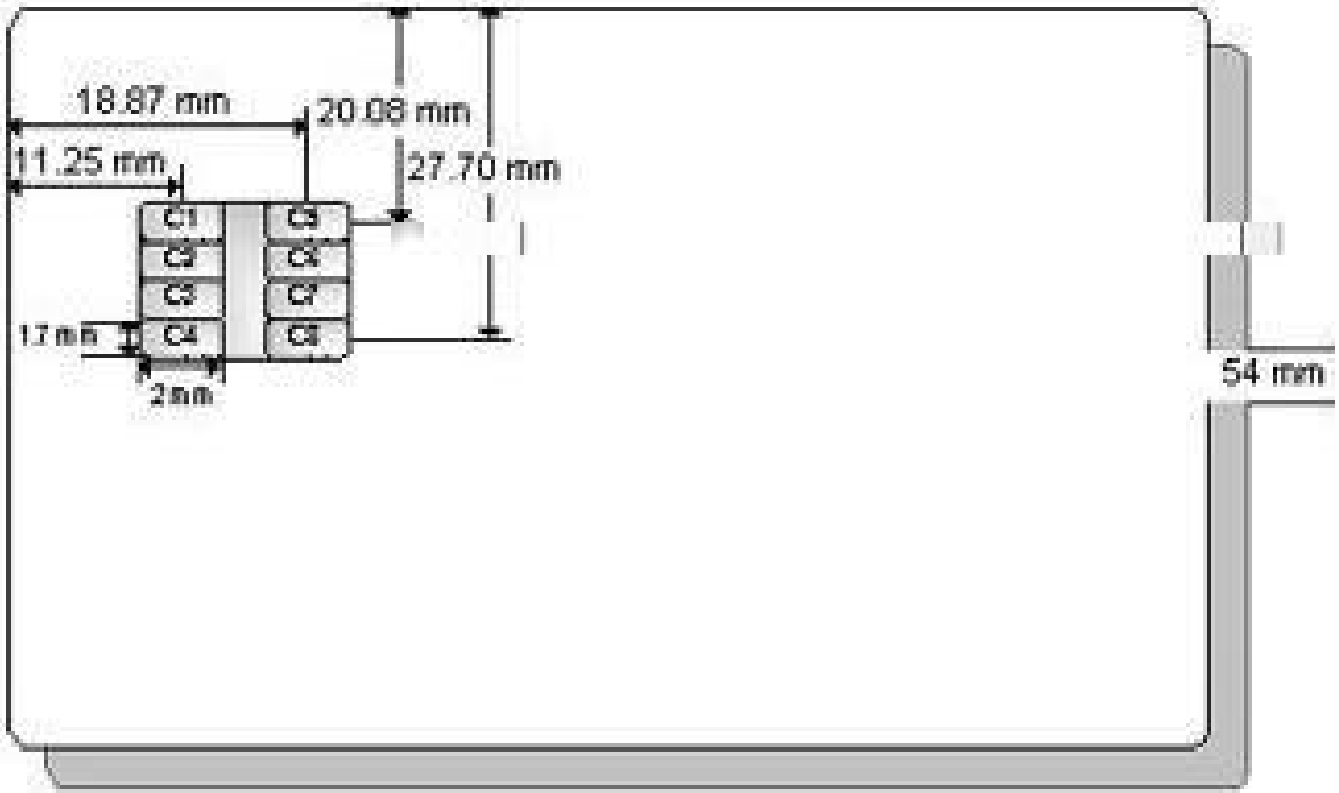
Forma unui smartcard

Un card are o formă dreptunghiulară de dimensiuni (*ISO*/*IEC* 7816) 85.60 × 53.98 mm, iar grosimea este 0.76 mm

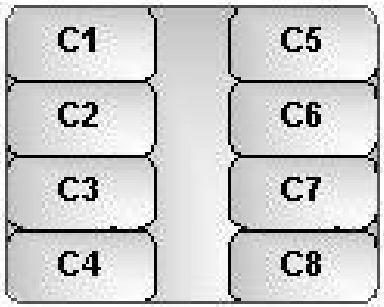
Pe una din fețe este inserat un micro-cip, de o anumită formă.
De exemplu:

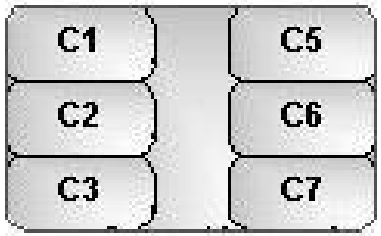


Oricare ar fi cip-ul inserat, el este poziționat astfel:

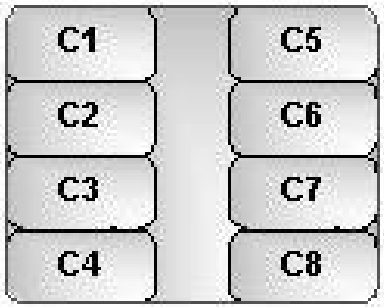


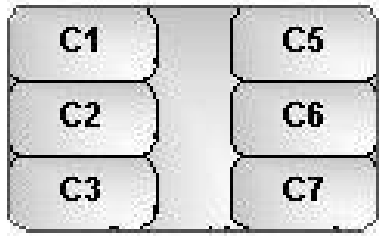
Detaliind, cele opt zone de contact ale unui cip sunt aranjate:





Detaliind, cele opt zone de contact ale unui cip sunt aranjate:





Poziție	Abreviere	Funcție
$C1$	VCC	Asigură voltajul
$C2$	RST	Reset
$C3$	CLK	Frecvență de ceas
$C4$	RFU	Rezervată pentru utilizare ulterioară
$C5$	GND	Înpământare
$C6$	VPP	Legătură pentru voltaj extern
$C7$	I/O	Legătură serială input/output
$C8$	RFU	Rezervată pentru utilizare ulterioară

Contactul *VPP*:

Este o reminiscență; a fost utilizat pentru a suplimenta voltajul la *EEPROM* în operațiile de programare și ștergere.

Din rațiuni de securitate, în prezent se face apel la el extrem de rar.

Contactul *VPP*:
Este o reminiscență; a fost utilizat pentru a suplimenta voltajul la *EEPROM* în operațiile de programare și ștergere.
Din rațiuni de securitate, în prezent se face apel la el extrem de rar.

Contactul *VPP* asigură un voltaj standard de 5 volți $\pm 10\%$.
Motivul este asigurarea unei compatibilități între piața smartcardurilor și cea a telefoniei mobile.

Contactul *VPP*:
Este o reminiscență; a fost utilizat pentru a suplimenta voltajul la *EEPROM* în operațiile de programare și ștergere.
Din rațiuni de securitate, în prezent se face apel la el extrem de rar.

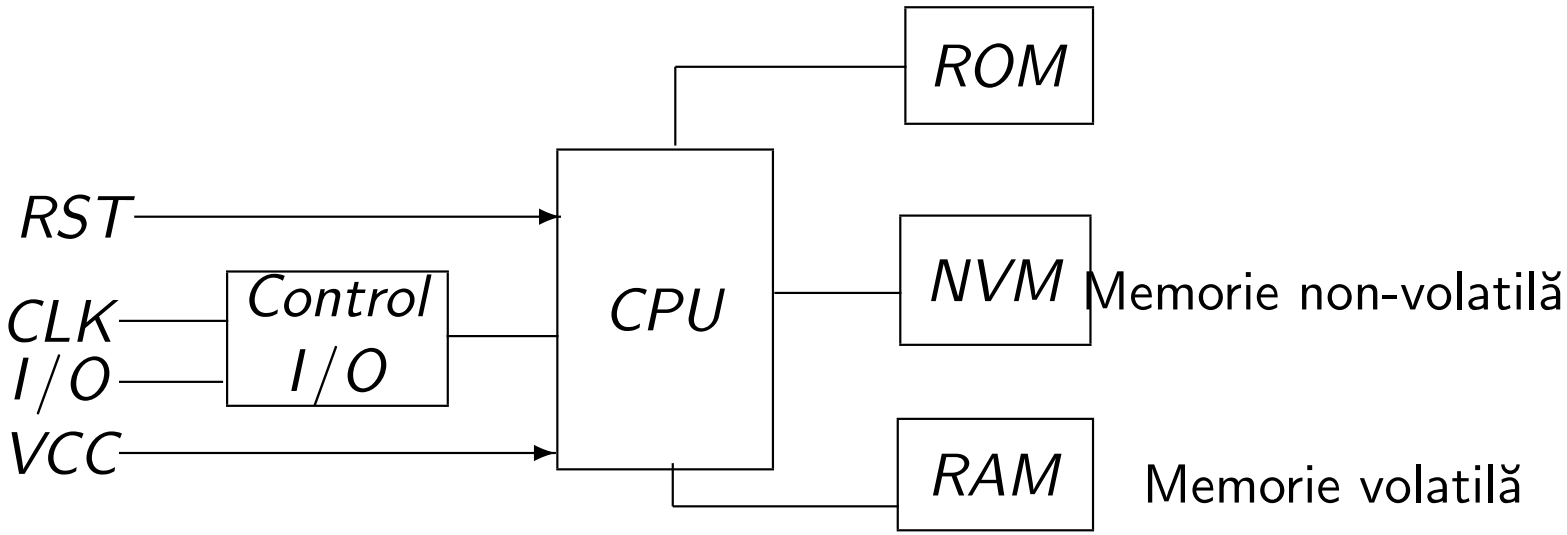
Contactul *VPP* asigură un voltaj standard de 5 volți $\pm 10\%$.
Motivul este asigurarea unei compatibilități între piața smartcardurilor și cea a telefoniei mobile.
Un telefon mobil lucrează pe o configurație de 3 volți, iar cardurile au rămas singura componentă care solicită telefonului mobil să aibă un convertor de încărcare.

Sistemul de calcul al unui smartcard

Este format dintr-un circuit integrat, care include unitatea centrală de procesare (*CPU*), memoria și liniile de Intrare/Ieșire (I/O).

Sistemul de calcul al unui smartcard

Este format dintr-un circuit integrat, care include unitatea centrală de procesare (*CPU*), memoria și liniile de Intrare/Ieșire (I/O).



Memoria

1. *Memoria read-only* (*ROM*).

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.
2. *Memoria non-volatilă* (*NVM*).

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.
2. *Memoria non-volatilă* (*NVM*).
   Poate reține informație chiar dacă sistemul de calcul este debranșat.

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.
2. *Memoria non-volatilă* (*NVM*).
   Poate reține informație chiar dacă sistemul de calcul este debranșat.
   Memoria non-volatilă este *EEPROM* (*Electrically Erasable Programmable Read-Only Memory*); conține date variabile (numerele de cont, sume de bani) care trebuie reținute de smartcard.

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.
2. *Memoria non-volatilă* (*NVM*).
   Poate reține informație chiar dacă sistemul de calcul este debranșat.
   Memoria non-volatilă este *EEPROM* (*Electrically Erasable Programmable Read-Only Memory*); conține date variabile (numerele de cont, sume de bani) care trebuie reținute de smartcard. O memorie *NVM* poate fi scrisă și citită de aplicații, dar nu poate fi folosită ca o memorie *RAM*.

Memoria

1. *Memoria read-only* (*ROM*).
   Conține sistemul de operare și rutine pentru comunicare și administrare a sistemului de fișiere.
2. *Memoria non-volatilă* (*NVM*).
   Poate reține informație chiar dacă sistemul de calcul este debranșat.
   Memoria non-volatilă este *EEPROM* (*Electrically Erasable Programmable Read-Only Memory*); conține date variabile (numerele de cont, sume de bani) care trebuie reținute de smartcard. O memorie *NVM* poate fi scrisă și citită de aplicații, dar nu poate fi folosită ca o memorie *RAM*.
3. Memorie *RAM*. Folosită de aplicațiile programatorului și de celelalte rutine utilitare.

Ieșirile/intrările unui smartcard

Canalul de intrare/ieșire este serial uni-direcțional.
Hardware-ul smartcardului poate suporta transfer de date de până la 115.200 biți/secundă, dar cititoarele de carduri comunică de obicei la viteze mult mai mici.

Ieșirile/intrările unui smartcard

Canalul de intrare/ieșire este serial uni-direcțional.
Hardware-ul smartcardului poate suporta transfer de date de până la 115.200 biți/secundă, dar cititoarele de carduri comunică de obicei la viteze mult mai mici.
Comunicarea dintre terminal și card se bazează pe o relație master (terminal) - slave (smartcard): terminalul trimite comenzi către card și așteaptă răspunsul.

Ieșirile/intrările unui smartcard

Canalul de intrare/ieșire este serial uni-direcțional.
Hardware-ul smartcardului poate suporta transfer de date de până la 115.200 biți/secundă, dar cititoarele de carduri comunică de obicei la viteze mult mai mici.
Comunicarea dintre terminal și card se bazează pe o relație master (terminal) - slave (smartcard): terminalul trimite comenzi către card și așteaptă răspunsul.
Smartcard-ul nu trimite niciodată date către terminal – decât sub forma unui răspuns la o comandă dată de terminal.

Software

Există două tipuri fundamentale de software pentru smartcard-uri:

Software

Există două tipuri fundamentale de software pentru smartcard-uri:

1. Software-ul pentru terminal (gazdă); aplicațiile înglobate în el rulează pe un computer conectat la un smartcard.

Software

Există două tipuri fundamentale de software pentru smartcard-uri:

1. Software-ul pentru terminal (gazdă); aplicațiile înglobate în el rulează pe un computer conectat la un smartcard.
2. Software-ul instalat pe card.

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;
2. Scoate date;

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;
2. Scoate date;
3. Citește date din memoria nonvolatilă (*NVM*);

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;
2. Scoate date;
3. Citește date din memoria nonvolatilă (*NVM*);
4. Scrie sau șterge date din *NVM*;

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;
2. Scoate date;
3. Citește date din memoria nonvolatilă (*NVM*);
4. Scrie sau șterge date din *NVM*;
5. Calculează o funcție criptografică.

Funcțiile unui smartcard

Un smartcard asigură cinci operații de bază:

1. Introduce date;
2. Scoate date;
3. Citește date din memoria nonvolatilă (*NVM*);
4. Scrie sau șterge date din *NVM*;
5. Calculează o funcție criptografică.

Gradul major de risc aparține operațiilor (2) și (4). Operația (2) oferă în exterior date și rezultate, iar (4) modifică conținutul *NVM*.

Cardul trebuie să fie sigur că operațiile sunt efectuate de proprietarul de card, sau că anumite comenzi primite au fost formulate de emițătorul de card.

Cardul trebuie să fie sigur că operațiile sunt efectuate de proprietarul de card, sau că anumite comenzi primite au fost formulate de emițătorul de card.
O serie de mecanisme și tehnici – de la *PIN* sau *MAC* până la scheme sofisticate de semnătură electronică – sunt utilizate de card pentru a controla aceste lucruri.
Aceste mecanisme sunt bazate atât pe tehnici criptografice cât folosind și tehnici empirice.

Cardul trebuie să fie sigur că operațiile sunt efectuate de proprietarul de card, sau că anumite comenzi primite au fost formulate de emițătorul de card.
O serie de mecanisme și tehnici – de la *PIN* sau *MAC* până la scheme sofisticate de semnătură electronică – sunt utilizate de card pentru a controla aceste lucruri.
Aceste mecanisme sunt bazate atât pe tehnici criptografice cât folosind și tehnici empirice.
Cardul poate reacționa la unele tipuri de încercări de fraudă. De exemplu, trei *PIN*-uri greșite vor duce la blocarea cardului.

Cardul trebuie să fie sigur că operațiile sunt efectuate de proprietarul de card, sau că anumite comenzi primite au fost formulate de emițătorul de card.
O serie de mecanisme și tehnici – de la *PIN* sau *MAC* până la scheme sofisticate de semnătură electronică – sunt utilizate de card pentru a controla aceste lucruri.
Aceste mecanisme sunt bazate atât pe tehnici criptografice cât folosind și tehnici empirice.
Cardul poate reacționa la unele tipuri de încercări de fraudă. De exemplu, trei *PIN*-uri greșite vor duce la blocarea cardului.
Creșterea dimensiunii memoriei și a vitezei de calcul asociate cu proceduri sofisticate de securitate fac posibilă elaborarea unui set sporit de mecanisme care să asigure securitatea logică.

Exemplu

Smartcardurile emise de unele unități bancare au memoria nonvolatilă segmentată (fizic) în mai multe zone:

Exemplu

Smartcardurile emise de unele unități bancare au memoria nonvolatilă segmentată (fizic) în mai multe zone:

- O zonă deschisă, accesibilă fără restricții;

Exemplu

Smartcardurile emise de unele unități bancare au memoria nonvolatilă segmentată (fizic) în mai multe zone:

- O zonă deschisă, accesibilă fără restricții;
- O zonă protejată unde citirea este liberă, dar pentru scriere este necesară cunoașterea unei parole;

Exemplu

Smartcardurile emise de unele unități bancare au memoria nonvolatilă segmentată (fizic) în mai multe zone:

- O zonă deschisă, accesibilă fără restricții;
- O zonă protejată unde citirea este liberă, dar pentru scriere este necesară cunoașterea unei parole;
- O zonă confidențială, unde accesul la citire este permis doar cu parolă;

Exemplu

Smartcardurile emise de unele unități bancare au memoria nonvolatilă segmentată (fizic) în mai multe zone:

- O zonă deschisă, accesibilă fără restricții;
- O zonă protejată unde citirea este liberă, dar pentru scriere este necesară cunoașterea unei parole;
- O zonă confidențială, unde accesul la citire este permis doar cu parolă;
- O zonă secretă care conține *PIN*-uri, parole și chei criptografice.

Sistemul de operare

Numit frecvent *COS* sau – uneori – *Mask*.

Sistemul de operare

Numit frecvent *COS* sau – uneori – *Mask*.
Este o secvență de instrucțiuni stocată în memoria *ROM* a smartcard-ului.

Sistemul de operare

Numit frecvent *COS* sau – uneori – *Mask*.
Este o secvență de instrucțiuni stocată în memoria *ROM* a smartcard-ului.
Sistemele de operare de pe un cip sunt împărțite în:

1. *COS* pentru scopuri generale; conțin un set de instrucțiuni generale care acoperă majoritatea aplicațiilor.

Sistemul de operare

Numit frecvent *COS* sau – uneori – *Mask*.
Este o secvență de instrucțiuni stocată în memoria *ROM* a smartcard-ului.
Sistemele de operare de pe un cip sunt împărțite în:

1. *COS* pentru scopuri generale; conțin un set de instrucțiuni generale care acoperă majoritatea aplicațiilor.
2. *COS*-uri dedicate; conțin comenzi speciale pentru unele aplicații – eventual însăși aplicația.

Sistemul de operare

Numit frecvent *COS* sau – uneori – *Mask*.
Este o secvență de instrucțiuni stocată în memoria *ROM* a smartcard-ului.
Sistemele de operare de pe un cip sunt împărțite în:

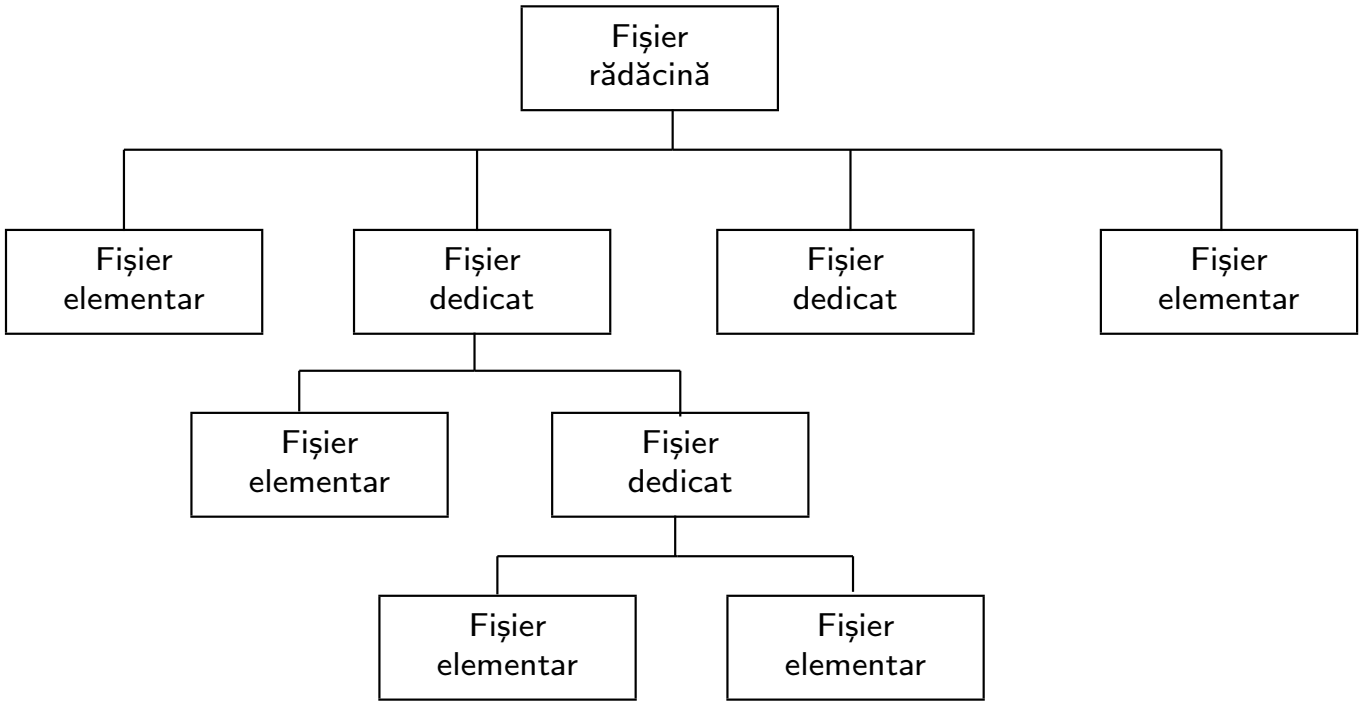
1. *COS* pentru scopuri generale; conțin un set de instrucțiuni generale care acoperă majoritatea aplicațiilor.
2. *COS*-uri dedicate; conțin comenzi speciale pentru unele aplicații – eventual însăși aplicația.
   Este cazul cardurilor special concepute în acest scop (mai ales în zona de comerț electronic).

Sistemul de fișiere al smartcardurilor

Deoarece smartcardurile nu au periferice, un sistem de fișiere smartcard are o singură rădăcină și o ierarhie bazată pe directoare.

Sistemul de fișiere al smartcardurilor

Deoarece smartcardurile nu au periferice, un sistem de fișiere smartcard are o singură rădăcină și o ierarhie bazată pe directoare.



Exemplu

Fișierele liniare: înregistrări de lungime fixă, care pot fi accesate după numărul înregistrării, sau citite secvențial folosind operațiile de citit următoarea sau precedenta înregistrare.

Exemplu

Fișierele liniare: înregistrări de lungime fixă, care pot fi accesate după numărul înregistrării, sau citite secvențial folosind operațiile de citit următoarea sau precedenta înregistrare.
Fișierele ciclice: fișiere liniare care ciclează înapoi la prima înregistrare atunci când după ultima înregistrare este citită sau scrisă încă o înregistrare.

Exemplu

Fișierele liniare: înregistrări de lungime fixă, care pot fi accesate după numărul înregistrării, sau citite secvențial folosind operațiile de citit următoarea sau precedenta înregistrare.
Fișierele ciclice: fișiere liniare care ciclează înapoi la prima înregistrare atunci când după ultima înregistrare este citită sau scrisă încă o înregistrare.
Fișierele "portofel": tip de fișier specific (folosit în aplicații de comerț electronic), suportat de sistemele de operare smartcard. Aceste fișiere sunt ciclice, fiecare înregistrare conținând informație despre o tranzacție a "portofelului" electronic.

Exemplu

Fișierele liniare: înregistrări de lungime fixă, care pot fi accesate după numărul înregistrării, sau citite secvențial folosind operațiile de citit următoarea sau precedenta înregistrare.
Fișierele ciclice: fișiere liniare care ciclează înapoi la prima înregistrare atunci când după ultima înregistrare este citită sau scrisă încă o înregistrare.
Fișierele "portofel": tip de fișier specific (folosit în aplicații de comerț electronic), suportat de sistemele de operare smartcard. Aceste fișiere sunt ciclice, fiecare înregistrare conținând informație despre o tranzacție a "portofelului" electronic.
Fișierele transparente: blocuri nediferențiate de memorie, care pot fi structurate pe aplicații în funcție de necesități.

APDU

Protocolul de bază în stabilirea unei linii de comunicație între smartcard și terminal este "*Application Protocol Data Units*" (*APDU*).

APDU

Protocolul de bază în stabilirea unei linii de comunicație între smartcard și terminal este "*Application Protocol Data Units*" (*APDU*).

Un *APDU* poate fi considerat un pachet de date care conține o cerere completă sau un răspuns complet dat de un smartcard.

APDU

Protocolul de bază în stabilirea unei linii de comunicație între smartcard și terminal este "*Application Protocol Data Units*" (*APDU*).

Un *APDU* poate fi considerat un pachet de date care conține o cerere completă sau un răspuns complet dat de un smartcard.

ISO 7816 – 4 definește două tipuri de *APDU* - uri.

De comandă:

Sunt trimise de aplicația gazdă.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.
2. *INS*: Octet care determină comanda respectivă.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.
2. *INS*: Octet care determină comanda respectivă.
3. *P*1, *P*2: Doi octeți pentru a transfera parametri specifici comenzii.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.
2. *INS*: Octet care determină comanda respectivă.
3. *P*1, *P*2: Doi octeți pentru a transfera parametri specifici comenzii.
4. *Lc*: Octet care codifică lungimea datelor opționale transmise card-ului cu acest *APDU*.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.
2. *INS*: Octet care determină comanda respectivă.
3. *P*1, *P*2: Doi octeți pentru a transfera parametri specifici comenzii.
4. *Lc*: Octet care codifică lungimea datelor opționale transmise card-ului cu acest *APDU*.
5. Date opționale: șirul de octeți – cu lungimea *Lc* – care poate conține datele propriu-zise ce vor fi procesate de card.

De comandă:

Sunt trimise de aplicația gazdă.Format:

CLA	*INS*	*P*1	*P*2	*Lc*	Date opționale	*Le*

1. *CLA*: Identifică clasa instrucțiunii.
2. *INS*: Octet care determină comanda respectivă.
3. *P*1, *P*2: Doi octeți pentru a transfera parametri specifici comenzii.
4. *Lc*: Octet care codifică lungimea datelor opționale transmise card-ului cu acest *APDU*.
5. Date opționale: șirul de octeți – cu lungimea *Lc* – care poate conține datele propriu-zise ce vor fi procesate de card.
6. *Le*: Octet care specifică lungimea datelor ce se așteaptă a fi returnate de către *APDU*-ul de răspuns următor.
   Dacă $Le = 00$, toate datele disponibile sunt trimise în răspunsul la comandă.

De răspuns:

Sunt trimise înapoi de smartcard, ca reacție la comenzi.

De răspuns:

Sunt trimise înapoi de smartcard, ca reacție la comenzi.
Formatul *APDU*-urilor de răspuns este:

Date opționale	*SW*1	*SW*2

De răspuns:

Sunt trimise înapoi de smartcard, ca reacție la comenzi.
Formatul *APDU*-urilor de răspuns este:

Date opționale	*SW*1	*SW*2

unde:

1. Date opționale: un șir de *Lc* octeți – dacă acest *APDU* este răspunsul unui *APDU* comandă ce avea octetul *Lc* setat pe o valoare nenulă.
   Altfel numărul octeților este variabil.

De răspuns:

Sunt trimise înapoi de smartcard, ca reacție la comenzi.
Formatul *APDU*-urilor de răspuns este:

Date opționale	*SW*1	*SW*2

unde:

1. Date opționale: un șir de *Lc* octeți – dacă acest *APDU* este răspunsul unui *APDU* comandă ce avea octetul *Lc* setat pe o valoare nenulă.
   Altfel numărul octeților este variabil.
2. *SW*1, *SW*2: Doi octeți de stare, care conțin informații de stare definite conform *ISO 7816-4*.

Aplicația software folosește un protocol bazat pe *APDU*-uri pentru schimbul de control și informație între card și cititorul de carduri.

Aplicația software folosește un protocol bazat pe *APDU*-uri pentru schimbul de control și informație între card și cititorul de carduri.

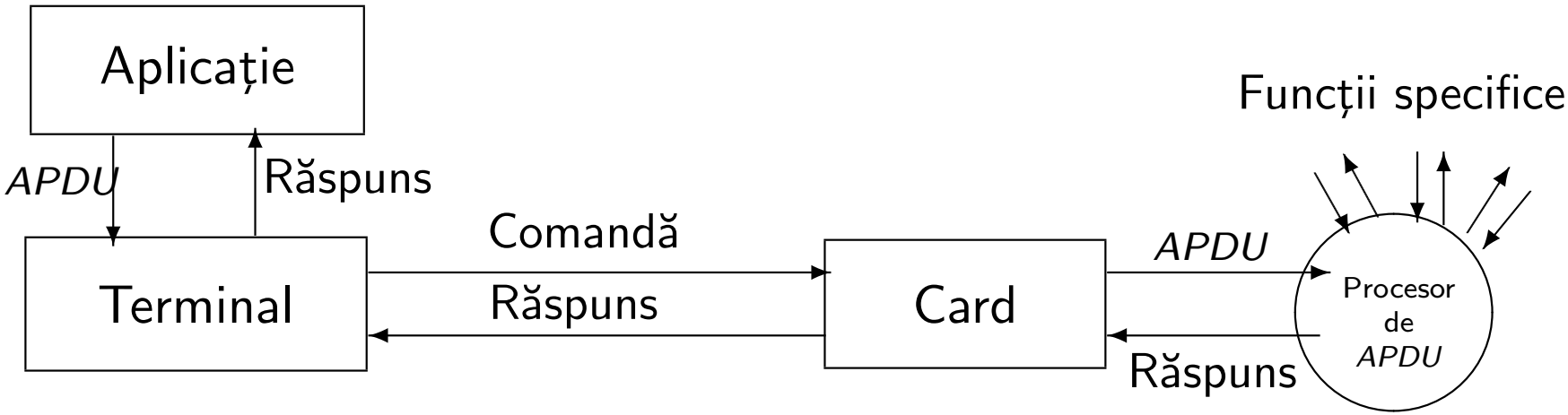
Aceste *APDU*-uri sunt schimbate folosindu-se protocoalele la nivel de conexiune $T = 0$ sau $T = 1$.
Protocolul $T = 0$ este orientat pe octeți, iar $T = 1$ este orientat pe blocuri de octeți.

Aplicația software folosește un protocol bazat pe *APDU*-uri pentru schimbul de control și informație între card și cititorul de carduri.

Aceste *APDU*-uri sunt schimbate folosindu-se protocoalele la nivel de conexiune $T = 0$ sau $T = 1$.
Protocolul $T = 0$ este orientat pe octeți, iar $T = 1$ este orientat pe blocuri de octeți.

Mai există și alte protocoale alternative pentru smartcardurile fără contacte (bazate pe tehnologia *JavaCard*) cum ar fi $T = USB$ sau $T = RF$.

O componentă software de pe card interpretează *APDU*-urile și execută operațiile specificate.



Protocolul $T = 0$ sau $T = 1$

Entități implicate la un smartcard

Există mai multe entități implicate potențial într-un sistem bazat pe smartcard.

Entități implicate la un smartcard

Există mai multe entități implicate potențial într-un sistem bazat pe smartcard.

Proprietarul cardului:
Persoana care posedă cardul în mod uzual; acesta decide dacă și când să-l folosească.

Entități implicate la un smartcard

Există mai multe entități implicate potențial într-un sistem bazat pe smartcard.

Proprietarul cardului:
Persoana care posedă cardul în mod uzual; acesta decide dacă și când să-l folosească.

Proprietarul poate controla datele de pe card oferite de sistem, dar extrem de rar are și controlul protocoalelor, softului sau hardului încorporate în card.

Proprietarul datelor

Este componenta care deține controlul datelor aflate pe card.

Proprietarul datelor

Este componenta care deține controlul datelor aflate pe card.

În anumite cazuri, el coincide cu proprietarul cardului.

Proprietarul datelor

Este componenta care deține controlul datelor aflate pe card.

În anumite cazuri, el coincide cu proprietarul cardului.

Pentru cardurile de plăți, proprietarul datelor este cel emite plățile cash; această diferențiere între cele două componente deschide anumite oportunități de atac.

Terminalul

Este componenta care asigură interacțiunea smartcardului cu exteriorul.

Terminalul

Este componenta care asigură interacțiunea smartcardului cu exteriorul.

Terminalul controlează toate intrările și ieșirile spre și dinspre smartcard: tastatura prin care datele sunt încorporate în smartcard și monitorul pe care este vizualizat smartcardul.

Terminalul

Este componenta care asigură interacțiunea smartcardului cu exteriorul.

Terminalul controlează toate intrările și ieșirile spre și dinspre smartcard: tastatura prin care datele sunt încorporate în smartcard și monitorul pe care este vizualizat smartcardul.

La un card telefonic, această componentă este chiar aparatul telefonic.
Pentru un card de plăți, terminalul este ATM-ul la care este făcută conexiunea.

Emițătorul de card

Este componenta care emite smartcardul.

Emițătorul de card

Este componenta care emite smartcardul.

El controlează sistemul de operare al smartcardului, precum și datele stocate inițial.

Emițătorul de card

Este componenta care emite smartcardul.

El controlează sistemul de operare al smartcardului, precum și datele stocate inițial.

Exemplu

- Pentru un card telefonic, această componentă este compania de telefoane.
- Pentru un card de serviciu, emițătorul este instituția la care este angajat proprietarul cardului.

Emițătorul de card

Este componenta care emite smartcardul.

El controlează sistemul de operare al smartcardului, precum și datele stocate inițial.

Exemplu

- Pentru un card telefonic, această componentă este compania de telefoane.
- Pentru un card de serviciu, emițătorul este instituția la care este angajat proprietarul cardului.

Din punct de vedere al securității este mai simplu de considerat că emițătorul de card, fabricantul și programatorul de software formează o singură componentă.
În realitate, acest lucru se întâmplă foarte rar.

Fabricantul de card

Este componenta care realizează smartcardul.

Fabricantul de card

Este componenta care realizează smartcardul.

Este o imagine simplificată; în realitate, fabricantul de card poate deține sau nu facilitățile de fabricare ale cipului încorporat.

Fabricantul de card

Este componenta care realizează smartcardul.

Este o imagine simplificată; în realitate, fabricantul de card poate deține sau nu facilitățile de fabricare ale cipului încorporat.

Exemplu

Există funcții subcontractate care folosesc componente externe sistemului (cum ar fi compilatoare VHDL); deși nu sunt în proprietatea fabricantului de card, ele sunt introduse pe card.

Programatorul de software

Este componenta care realizează programele rezidente pe smartcard.

Programatorul de software

Este componenta care realizează programele rezidente pe smartcard.

Și aici problemele sunt mult simplificate: în realitate sunt mai mulți programatori care scriu codul pentru compilatoare, programe utilitare, componente de securitate etc.

Tipuri de smartcard

- Cardul de plăți (Digital Stored Value Card): Folosit pentru înlocuirea modului de plată cash. Exemplu: Both Mondex, VisaCash.

Tipuri de smartcard

- Cardul de plăți (Digital Stored Value Card): Folosit pentru înlocuirea modului de plată cash. Exemplu: Both Mondex, VisaCash.

  Aici comerciantul este proprietarul terminalului, iar cumpărătorul este proprietarul de card.

  Atât propietarul datelor cât și emițătorul de card sunt instituția financiară (de obicei o bancă) care sprijină sistemul.

Tipuri de smartcard

- Cardul de plăți (Digital Stored Value Card): Folosit pentru înlocuirea modului de plată cash. Exemplu: Both Mondex, VisaCash.

  Aici comerciantul este proprietarul terminalului, iar cumpărătorul este proprietarul de card.

  Atât propietarul datelor cât și emițătorul de card sunt instituția financiară (de obicei o bancă) care sprijină sistemul.
- Cardul digital (Digital Check Card): Similar cardului de plăți, cu diferența că proprietarul cardului este același cu proprietarul datelor.

Tipuri de smartcard

- Cardul de plăți (Digital Stored Value Card): Folosit pentru înlocuirea modului de plată cash. Exemplu: Both Mondex, VisaCash.

  Aici comerciantul este proprietarul terminalului, iar cumpărătorul este proprietarul de card.

  Atât propietarul datelor cât și emițătorul de card sunt instituția financiară (de obicei o bancă) care sprijină sistemul.
- Cardul digital (Digital Check Card): Similar cardului de plăți, cu diferența că proprietarul cardului este același cu proprietarul datelor.
- Cartelă telefonică Prepaid: Card de plăți cu utilizare specificată. Proprietarul cardului este utilizatorul.

  Terminalul, datele și emițătorul cardului sunt asigurate de o singură entitate: compania de telefoane.

Tipuri de smartcard

- Cartelă telefonică: Smartcardul stochează doar un număr de cont – un pointer la o bază de date.

  Proprietarul cardului și al datelor este utilizatorul, iar cel care emite cardul, deținând și terminalul, este compania de telefoane.

Tipuri de smartcard

- Cartelă telefonică: Smartcardul stochează doar un număr de cont – un pointer la o bază de date.

  Proprietarul cardului și al datelor este utilizatorul, iar cel care emite cardul, deținând și terminalul, este compania de telefoane.

- Card de acces (Access Token): Smartcardul stochează o cheie care este utilizată într-un login sau protocol de autentificare.

  Pentru o instituție, proprietarul cardului este angajatul, iar proprietarul datelor, al terminalului și emițătorul de card este compania.

Tipuri de smartcard

- Cartelă telefonică: Smartcardul stochează doar un număr de cont – un pointer la o bază de date.

  Proprietarul cardului și al datelor este utilizatorul, iar cel care emite cardul, deținând și terminalul, este compania de telefoane.

- Card de acces (Access Token): Smartcardul stochează o cheie care este utilizată într-un login sau protocol de autentificare.

  Pentru o instituție, proprietarul cardului este angajatul, iar proprietarul datelor, al terminalului și emițătorul de card este compania.

  În cazul unui card de acces multifuncțional, proprietarul cardului și al datelor poate fi eventual una și aceeași persoană, în timp ce terminalul este deținut de un organism comercial, iar emițătorul de card – de o instituție financiară.

Tipuri de smartcard

- Card de navigare Web (Web Browsing Card): Utilizatorul poate folosi cardul pe propriul PC pentru a cumpăra diverse produse oferite prin Internet.

  Este o variantă a cardului de plăți, cu diferența că cel care face cumpărături deține atât cardul cât și terminalul.

Tipuri de smartcard

- Card de navigare Web (Web Browsing Card): Utilizatorul poate folosi cardul pe propriul PC pentru a cumpăra diverse produse oferite prin Internet.

  Este o variantă a cardului de plăți, cu diferența că cel care face cumpărături deține atât cardul cât și terminalul.
- Card de certificare (Digital Credential Device): Smartcardul stochează certificate digitale sau alte credențiale necesare autentificării față de parteneri.

Tipuri de smartcard

- Card de navigare Web (Web Browsing Card): Utilizatorul poate folosi cardul pe propriul PC pentru a cumpăra diverse produse oferite prin Internet.

  Este o variantă a cardului de plăți, cu diferența că cel care face cumpărături deține atât cardul cât și terminalul.
- Card de certificare (Digital Credential Device): Smartcardul stochează certificate digitale sau alte credențiale necesare autentificării față de parteneri.Proprietarul cardului este și proprietarul datelor. Emițătorul cardului este *CA*-ul care emite credențialele, sau o companie care le gestionează.

Tipuri de smartcard

- Card de navigare Web (Web Browsing Card): Utilizatorul poate folosi cardul pe propriul PC pentru a cumpăra diverse produse oferite prin Internet.

  Este o variantă a cardului de plăți, cu diferența că cel care face cumpărături deține atât cardul cât și terminalul.
- Card de certificare (Digital Credential Device): Smartcardul stochează certificate digitale sau alte credențiale necesare autentificării față de parteneri.Proprietarul cardului este și proprietarul datelor. Emițătorul cardului este *CA*-ul care emite credențialele, sau o companie care le gestionează.
- Cartelă de stocare a cheilor (Key Storage Card): Utilizatorul stochează diverse chei publice pe smartcard, pentru a le păstra într-un mediu mai sigur decât propriul PC.

Tipuri de smartcard

- Card de navigare Web (Web Browsing Card): Utilizatorul poate folosi cardul pe propriul PC pentru a cumpăra diverse produse oferite prin Internet.

  Este o variantă a cardului de plăți, cu diferența că cel care face cumpărături deține atât cardul cât și terminalul.
- Card de certificare (Digital Credential Device): Smartcardul stochează certificate digitale sau alte credențiale necesare autentificării față de parteneri.Proprietarul cardului este și proprietarul datelor. Emițătorul cardului este *CA*-ul care emite credențialele, sau o companie care le gestionează.
- Cartelă de stocare a cheilor (Key Storage Card): Utilizatorul stochează diverse chei publice pe smartcard, pentru a le păstra într-un mediu mai sigur decât propriul PC.
  El este proprietarul cardului, datelor, și terminalului.

Componente criptografice integrate pe un smartcard

Smartcard-urile conțin suficient de multe componente criptografice pe care se dezvoltă aplicații și protocoale de securitate cunoscute.

Componente criptografice integrate pe un smartcard

Smartcard-urile conțin suficient de multe componente criptografice pe care se dezvoltă aplicații și protocoale de securitate cunoscute.

- *Semnătura electronică:*
  Cele mai folosite protocoale de semnătură sunt bazate pe *RSA*; pe ele sunt implementate chei de 512, 768 sau 1024 biți. Timpul necesar unei semnături digitale este de obicei sub o secundă.

Componente criptografice integrate pe un smartcard

Smartcard-urile conțin suficient de multe componente criptografice pe care se dezvoltă aplicații și protocoale de securitate cunoscute.

- *Semnătura electronică:*
  Cele mai folosite protocoale de semnătură sunt bazate pe *RSA*; pe ele sunt implementate chei de 512, 768 sau 1024 biți. Timpul necesar unei semnături digitale este de obicei sub o secundă. Fișierul *EEPROM* care conține cheia privată este conceput astfel încât informațiile sensibile despre cheie să nu părăsească niciodată cip-ul. Folosirea cheii private este protejată de *PIN*-ul utilizatorului.

Componente criptografice integrate pe un smartcard

Smartcard-urile conțin suficient de multe componente criptografice pe care se dezvoltă aplicații și protocoale de securitate cunoscute.

- *Semnătura electronică:*
  Cele mai folosite protocoale de semnătură sunt bazate pe *RSA*; pe ele sunt implementate chei de 512, 768 sau 1024 biți. Timpul necesar unei semnături digitale este de obicei sub o secundă. Fișierul *EEPROM* care conține cheia privată este conceput astfel încât informațiile sensibile despre cheie să nu părăsească niciodată cip-ul. Folosirea cheii private este protejată de *PIN*-ul utilizatorului.
  Observație: Pentru generarea unei perechi de chei *RSA* de 1024 biți se poate aștepta și 3 minute; de aceea este necesară existența unui hard specializat sau a unui co-procesor dedicat.

Componente criptografice integrate pe un smartcard

Smartcard-urile conțin suficient de multe componente criptografice pe care se dezvoltă aplicații și protocoale de securitate cunoscute.

- *Semnătura electronică:*
  Cele mai folosite protocoale de semnătură sunt bazate pe *RSA*; pe ele sunt implementate chei de 512, 768 sau 1024 biți. Timpul necesar unei semnături digitale este de obicei sub o secundă.
  Fișierul *EEPROM* care conține cheia privată este conceput astfel încât informațiile sensibile despre cheie să nu părăsească niciodată cip-ul. Folosirea cheii private este protejată de *PIN*-ul utilizatorului.
  Observație: Pentru generarea unei perechi de chei *RSA* de 1024 biți se poate aștepta și 3 minute; de aceea este necesară existența unui hard specializat sau a unui co-procesor dedicat.
  De asemenea, lipsa puterii de procesare implică implementarea de generatori relativ slabi de numere pseudo-aleatoare.

Algoritmi de criptare

Algoritmul de Semnătură Digitală (*DSA*) este implementat mai puțin frecvent decât RSA. Atunci când este disponibil, el folosește chei de numai 512 biți.

DES și 3*DES* sunt întâlnite frecvent în smartcard-urile performante.

Algoritmi de criptare

Algoritmul de Semnătură Digitală (*DSA*) este implementat mai puțin frecvent decât RSA. Atunci când este disponibil, el folosește chei de numai 512 biți.

DES și 3*DES* sunt întâlnite frecvent în smartcard-urile performante.
De obicei există și opțiunea de a fi folosiți în calculul unui *MAC*.

Algoritmi de criptare

Algoritmul de Semnătură Digitală (*DSA*) este implementat mai puțin frecvent decât RSA. Atunci când este disponibil, el folosește chei de numai 512 biți.

DES și 3*DES* sunt întâlnite frecvent în smartcard-urile performante.
De obicei există și opțiunea de a fi folosiți în calculul unui *MAC*.

Datorită faptului că interfața serială a smartcard-ului are o lățime mică de bandă, criptarea cu un sistem simetric a unei cantități mari de informație este foarte lentă.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.
- Algoritmii de dispersie cuprind de obicei $SHA-1$ și $MD5$.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.
- Algoritmii de dispersie cuprind de obicei $SHA-1$ și $MD5$. Lățimea scăzută a benzii seriale de comunicare împiedică folosirea lor pentru dispersia unui volum mare de informații.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.
- Algoritmii de dispersie cuprind de obicei $SHA-1$ și $MD5$. Lățimea scăzută a benzii seriale de comunicare împiedică folosirea lor pentru dispersia unui volum mare de informații.
- Generatoarele de numere pseudo-aleatoare (RNG) diferă de la un tip de card la altul.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.
- Algoritmii de dispersie cuprind de obicei $SHA-1$ și $MD5$. Lățimea scăzută a benzii seriale de comunicare împiedică folosirea lor pentru dispersia unui volum mare de informații.
- Generatoarele de numere pseudo-aleatoare (RNG) diferă de la un tip de card la altul.De exemplu se implementează un pseudo-RNG în care fiecare card are propria sa inițializare. În acest caz, numerele aleatoare ciclează în funcție de algoritm și inițializare.

Alte aplicații

- Funcția de portofel electronic este prezentă în multe aplicații; se bazează pe criptări cu cheie simetrică.
- Algoritmii de dispersie cuprind de obicei $SHA-1$ și $MD5$. Lățimea scăzută a benzii seriale de comunicare împiedică folosirea lor pentru dispersia unui volum mare de informații.
- Generatoarele de numere pseudo-aleatoare (RNG) diferă de la un tip de card la altul.De exemplu se implementează un pseudo-RNG în care fiecare card are propria sa inițializare. În acest caz, numerele aleatoare ciclează în funcție de algoritm și inițializare.

În smartcard-urile actuale sunt prezente diferite metode de monitorizare a securității hardware.

- O măsură de siguranță (ireversibilă) dezactivează orice cod de test din *EEPROM*.

În smartcard-urile actuale sunt prezente diferite metode de monitorizare a securității hardware.

- O măsură de siguranță (ireversibilă) dezactivează orice cod de test din *EEPROM*.Ea se poate folosi o singură dată în perioada de viață a unui smartcard.

În smartcard-urile actuale sunt prezente diferite metode de monitorizare a securității hardware.

- O măsură de siguranță (ireversibilă) dezactivează orice cod de test din *EEPROM*.Ea se poate folosi o singură dată în perioada de viață a unui smartcard.
- Pentru a evita clonarea cardurilor, o secvență nealterabilă specifică este parcursă obligatoriu la orice utilizare a memoriei *ROM*.

În smartcard-urile actuale sunt prezente diferite metode de monitorizare a securității hardware.

- O măsură de siguranță (ireversibilă) dezactivează orice cod de test din *EEPROM*.Ea se poate folosi o singură dată în perioada de viață a unui smartcard.
- Pentru a evita clonarea cardurilor, o secvență nealterabilă specifică este parcursă obligatoriu la orice utilizare a memoriei *ROM*.
- Când detectează fluctuații de voltaj, temperatură sau frecvență de ceas, cardurile sunt concepute să se reseteze automat într-o stare inițială.

În smartcard-urile actuale sunt prezente diferite metode de monitorizare a securității hardware.

- O măsură de siguranță (ireversibilă) dezactivează orice cod de test din *EEPROM*.Ea se poate folosi o singură dată în perioada de viață a unui smartcard.
- Pentru a evita clonarea cardurilor, o secvență nealterabilă specifică este parcursă obligatoriu la orice utilizare a memoriei *ROM*.
- Când detectează fluctuații de voltaj, temperatură sau frecvență de ceas, cardurile sunt concepute să se reseteze automat într-o stare inițială.
- Operația de citire din exterior a memoriei *ROM* este de obicei dezactivată.

Protocoalele de comunicare ale smartcard-urilor la nivel de comandă includ și un protocol de securitate.
Acesta este bazat de obicei pe tehnologii cu cheie simetrică și oferă smartcard-ului posibilitatea să autentifice terminalul cu care intră în contact direct.

Protocoalele de comunicare ale smartcard-urilor la nivel de comandă includ și un protocol de securitate.
Acesta este bazat de obicei pe tehnologii cu cheie simetrică și oferă smartcard-ului posibilitatea să autentifice terminalul cu care intră în contact direct.

Smartcard-urile au abilitatea de a configura mai multe *PIN*-uri cu scopuri diferite.

Protocoalele de comunicare ale smartcard-urilor la nivel de comandă includ și un protocol de securitate.
Acesta este bazat de obicei pe tehnologii cu cheie simetrică și oferă smartcard-ului posibilitatea să autentifice terminalul cu care intră în contact direct.

Smartcard-urile au abilitatea de a configura mai multe *PIN*-uri cu scopuri diferite.
Aplicațiile pot configura un *PIN* pentru supervizor, care să poată debloca *PIN*-ul unui utilizator – după un număr de încercări eșuate – sau să reinițializeze cardul.

Protocoalele de comunicare ale smartcard-urilor la nivel de comandă includ și un protocol de securitate.
Acesta este bazat de obicei pe tehnologii cu cheie simetrică și oferă smartcard-ului posibilitatea să autentifice terminalul cu care intră în contact direct.

Smartcard-urile au abilitatea de a configura mai multe *PIN*-uri cu scopuri diferite.
Aplicațiile pot configura un *PIN* pentru supervizor, care să poată debloca *PIN*-ul unui utilizator – după un număr de încercări eșuate – sau să reinițializeze cardul.
Alte *PIN*-uri pot fi configurate pentru a controla accesul la fișiere sensibile sau funcții ale portofelului electronic.

Codul *PIN*

Cea mai simplă metodă de identificare a unui utilizator este de a-i atașa un număr secret, numit *Personal Identification Number*.

Codul *PIN*

Cea mai simplă metodă de identificare a unui utilizator este de a-i atașa un număr secret, numit *Personal Identification Number*.

Un *PIN* este un număr de 4 cifre zecimale.
El este introdus folosind un terminal sau o tastatură de calculator și direcționat spre smartcard.

Cardul compară valoarea primită cu cea stocată în interior și reportează spre terminal rezultatul comparării.

Un *PIN* poate fi de două feluri: static sau modificabil.

- Un *PIN static* nu poate fi schimbat de utilizator.

Un *PIN* poate fi de două feluri: static sau modificabil.

- Un *PIN static* nu poate fi schimbat de utilizator.
  Dacă devine public, utilizatorul trebuie să distrugă cardul și să obțină un card nou, cu alt *PIN*.

Un *PIN* poate fi de două feluri: static sau modificabil.

- Un *PIN static* nu poate fi schimbat de utilizator.
  Dacă devine public, utilizatorul trebuie să distrugă cardul și să obțină un card nou, cu alt *PIN*.
- Un *PIN modificabil* poate fi modificat la solicitarea utilizatorului sau schimbat de acesta cu un număr mai ușor de memorat.

Un *PIN* poate fi de două feluri: static sau modificabil.

- Un *PIN static* nu poate fi schimbat de utilizator.
  Dacă devine public, utilizatorul trebuie să distrugă cardul și să obțină un card nou, cu alt *PIN*.
- Un *PIN modificabil* poate fi modificat la solicitarea utilizatorului sau schimbat de acesta cu un număr mai ușor de memorat.
  Risc: utilizatorul preferă adesea un *PIN* de forma $'1234'$, sau legat de date personale (data nașterii, căsătoriei etc).
  Smartcardul nu poate controla alegeri banale, deoarece nu are suficientă memorie pentru a construi o tabelă cu ele.
  În schimb un terminal poate interzice introducerea unor numere de forma $'1234'$, $'9999'$,$'1111'$, $'0000'$ etc.

Există chei personale de deblocare (*PUK*), numite și *super-PIN*.

Există chei personale de deblocare (*PUK*), numite și *super-PIN*. Ele au mai mult de 4 cifre (6 cifre de obicei) și sunt folosite pentru resetarea counterului unui smartcard atunci numărul de încercări a fost depășit.
Odată cu utilizarea unui *PUK* se va defini un nou *PIN*, deoarece un număr mare de încercări cu *PIN*-uri greșite este un indiciu – în variantă optimistă – că utilizatorul a uitat propriul *PIN*.

Există chei personale de deblocare (*PUK*), numite și *super-PIN*.
Ele au mai mult de 4 cifre (6 cifre de obicei) și sunt folosite pentru resetarea counterului unui smartcard atunci numărul de încercări a fost depășit.
Odată cu utilizarea unui *PUK* se va defini un nou *PIN*, deoarece un număr mare de încercări cu *PIN*-uri greșite este un indiciu – în variantă optimistă – că utilizatorul a uitat propriul *PIN*.

Alte aplicații folosesc *PIN*-uri "de transport".

Există chei personale de deblocare (*PUK*), numite și *super-PIN*.
Ele au mai mult de 4 cifre (6 cifre de obicei) și sunt folosite pentru resetarea counterului unui smartcard atunci numărul de încercări a fost depășit.
Odată cu utilizarea unui *PUK* se va defini un nou *PIN*, deoarece un număr mare de încercări cu *PIN*-uri greșite este un indiciu – în variantă optimistă – că utilizatorul a uitat propriul *PIN*.

Alte aplicații folosesc *PIN*-uri "de transport".
Smartcardul este personalizat cu un *PIN* generat aleator, pe care proprietarul de card îl primește într-o scrisoare trimisă prin poștă.
În momentul folosirii pentru prima oară a cardului, el înlocuiește acest *PIN* cu unul propriu.

Există chei personale de deblocare (*PUK*), numite și *super-PIN*.
Ele au mai mult de 4 cifre (6 cifre de obicei) și sunt folosite pentru resetarea counterului unui smartcard atunci numărul de încercări a fost depășit.
Odată cu utilizarea unui *PUK* se va defini un nou *PIN*, deoarece un număr mare de încercări cu *PIN*-uri greșite este un indiciu – în variantă optimistă – că utilizatorul a uitat propriul *PIN*.

Alte aplicații folosesc *PIN*-uri "de transport".
Smartcardul este personalizat cu un *PIN* generat aleator, pe care proprietarul de card îl primește într-o scrisoare trimisă prin poștă.
În momentul folosirii pentru prima oară a cardului, el înlocuiește acest *PIN* cu unul propriu.
O metodă similară este *PIN-zero*: cardul este preîncărcat cu un *PIN* banal, cum ar fi $'0000'$, valoare care este înlocuită obligatoriu la prima utilizare a cardului.

Tipuri de atac contra smartcardurilor

Definiție

Un atac este o încercare a uneia sau mai multor părți – implicate într-o tranzacție bazată pe smartcard – de a trișa.

Tipuri de atac contra smartcardurilor

Definiție

Un atac este o încercare a uneia sau mai multor părți – implicate într-o tranzacție bazată pe smartcard – de a trișa.

Sunt două tipuri de atac:

- Atacuri venite din partea componentelor sistemului de smartcard. De exemplu: încercări ale proprietarului de card de a înșela terminalul; sau – ale fabricantului de card de a înșela proprietarul de card.

Tipuri de atac contra smartcardurilor

Definiție

Un atac este o încercare a uneia sau mai multor părți – implicate într-o tranzacție bazată pe smartcard – de a trișa.

Sunt două tipuri de atac:

- Atacuri venite din partea componentelor sistemului de smartcard. De exemplu: încercări ale proprietarului de card de a înșela terminalul; sau – ale fabricantului de card de a înșela proprietarul de card.
- Atacuri venite dinafara sistemului. Sunt generate de obicei de persoane care fură un card de la proprietar sau care înlocuiesc softul/hardul terminalului.

Atacuri ale terminalului contra proprietarului de card

Realizat de obicei cu ATM-uri contrafăcute.

Atacuri ale terminalului contra proprietarului de card

Realizat de obicei cu ATM-uri contrafăcute.
Când un card este introdus într-un terminal, proprietarul său are încredere că acesta va lucra corect.
Mecanismele de eliminare a acestui atac se bazează pe faptul că pentru un interval scurt de timp, doar terminalul are acces la card.
De exemplu, multe softuri de pe card limitează suma ce poate fi retrasă la o cerere, sau într-o perioadă de timp.

Atacuri ale terminalului contra proprietarului de card

Realizat de obicei cu ATM-uri contrafăcute.
Când un card este introdus într-un terminal, proprietarul său are încredere că acesta va lucra corect.
Mecanismele de eliminare a acestui atac se bazează pe faptul că pentru un interval scurt de timp, doar terminalul are acces la card.
De exemplu, multe softuri de pe card limitează suma ce poate fi retrasă la o cerere, sau într-o perioadă de timp.
Aceste mecanisme nu semnalează nici o anomalie a terminalului; încearcă doar limitarea unor eventuale pagube.

Atacuri ale terminalului contra proprietarului de card

Realizat de obicei cu ATM-uri contrafăcute.
Când un card este introdus într-un terminal, proprietarul său are încredere că acesta va lucra corect.
Mecanismele de eliminare a acestui atac se bazează pe faptul că pentru un interval scurt de timp, doar terminalul are acces la card.
De exemplu, multe softuri de pe card limitează suma ce poate fi retrasă la o cerere, sau într-o perioadă de timp.
Aceste mecanisme nu semnalează nici o anomalie a terminalului; încearcă doar limitarea unor eventuale pagube.
Există și mecanisme de prevenire împotriva acestui tip de atac.
Exemplu: sisteme centrale care monitorizează toate cardurile și terminalele și semnalează orice comportament neadecvat.

Atacuri ale proprietarului de card contra terminalului

Folosesc carduri false sau modificate, bazate pe softuri pirat, cu intenția de a păcăli protocolul de autentificare card - terminal.

Atacuri ale proprietarului de card contra terminalului

Folosesc carduri false sau modificate, bazate pe softuri pirat, cu intenția de a păcăli protocolul de autentificare card - terminal.

De obicei un protocol bun reduce la minimum acest tip de atac. În plus, pot fi utilizate pentru autentificare anumite aspecte fizice ale cardului, greu de imitat (de exemplu hologramele utilizate de sistemele Visa și Mastercard) și care sunt controlate separat de terminal.

Atacuri ale proprietarului de card contra terminalului

Folosesc carduri false sau modificate, bazate pe softuri pirat, cu intenția de a păcăli protocolul de autentificare card - terminal.

De obicei un protocol bun reduce la minimum acest tip de atac. În plus, pot fi utilizate pentru autentificare anumite aspecte fizice ale cardului, greu de imitat (de exemplu hologramele utilizate de sistemele Visa și Mastercard) și care sunt controlate separat de terminal.
De remarcat că o semnătură digitală controlată de software nu este sigură, deoarece un card fals poate contraface (adesea) o semnătură, iar terminalul nu are nici o posibilitate de a inspecta interiorul cardului.

Atacuri ale proprietarului de card contra terminalului

Folosesc carduri false sau modificate, bazate pe softuri pirat, cu intenția de a păcăli protocolul de autentificare card - terminal.

De obicei un protocol bun reduce la minimum acest tip de atac. În plus, pot fi utilizate pentru autentificare anumite aspecte fizice ale cardului, greu de imitat (de exemplu hologramele utilizate de sistemele Visa și Mastercard) și care sunt controlate separat de terminal.
De remarcat că o semnătură digitală controlată de software nu este sigură, deoarece un card fals poate contraface (adesea) o semnătură, iar terminalul nu are nici o posibilitate de a inspecta interiorul cardului.

Apărarea constă în definirea unor funcții care să nu permită accesul proprietarului de card la datele din interiorul cardului.

Proprietarul de card contra proprietarul de date

În multe smartcarduri (mai ales cele folosite în sisteme comerciale), datele stocate pe card trebuie protejate față de proprietarul de card.

Proprietarul de card contra proprietarul de date

În multe smartcarduri (mai ales cele folosite în sisteme comerciale), datele stocate pe card trebuie protejate față de proprietarul de card.

Exemplu

O cartelă de acces într-o clădire poate conține o valoare secretă; cunoașterea ei va permite proprietarului să construiască un acces multiplu (pe mai multe zone ale clădirii).

Proprietarul de card contra proprietarul de date

În multe smartcarduri (mai ales cele folosite în sisteme comerciale), datele stocate pe card trebuie protejate față de proprietarul de card.

Exemplu

O cartelă de acces într-o clădire poate conține o valoare secretă; cunoașterea ei va permite proprietarului să construiască un acces multiplu (pe mai multe zone ale clădirii).

Cunoașterea unei chei secrete de pe un card de comerț electronic poate permite proprietarului să efectueze tranzacții ilegale.

Proprietarul de card contra proprietarul de date

În multe smartcarduri (mai ales cele folosite în sisteme comerciale), datele stocate pe card trebuie protejate față de proprietarul de card.

Exemplu

O cartelă de acces într-o clădire poate conține o valoare secretă; cunoașterea ei va permite proprietarului să construiască un acces multiplu (pe mai multe zone ale clădirii).

Cunoașterea unei chei secrete de pe un card de comerț electronic poate permite proprietarului să efectueze tranzacții ilegale.

În alte cazuri, se permite cunoașterea valorii de către proprietarul cardului, dar nu și modificarea ei.

Proprietarul de card contra proprietarul de date

În multe smartcarduri (mai ales cele folosite în sisteme comerciale), datele stocate pe card trebuie protejate față de proprietarul de card.

Exemplu

O cartelă de acces într-o clădire poate conține o valoare secretă; cunoașterea ei va permite proprietarului să construiască un acces multiplu (pe mai multe zone ale clădirii).

Cunoașterea unei chei secrete de pe un card de comerț electronic poate permite proprietarului să efectueze tranzacții ilegale.

În alte cazuri, se permite cunoașterea valorii de către proprietarul cardului, dar nu și modificarea ei.

Este cazul unui card de debit: dacă proprietarul poate modifica valoarea, el poate scoate bani pe care nu îi are în cont.

Proprietăți

- Cardul acționează ca un perimetru de securitate, limitând strict accesul proprietarului la datele aflate în interiorul cardului.

Proprietăți

- Cardul acționează ca un perimetru de securitate, limitând strict accesul proprietarului la datele aflate în interiorul cardului.
- Atacatorul are acces la card în condiții stabilite de el: poate să-l supună oricăror verificări și experimente dorește, pentru a avea acces la informațiile stocate.

Proprietăți

- Cardul acționează ca un perimetru de securitate, limitând strict accesul proprietarului la datele aflate în interiorul cardului.
- Atacatorul are acces la card în condiții stabilite de el: poate să-l supună oricăror verificări și experimente dorește, pentru a avea acces la informațiile stocate.
  În particular, poate distruge cardul, pentru a vedea cum este construit.

Proprietăți

- Cardul acționează ca un perimetru de securitate, limitând strict accesul proprietarului la datele aflate în interiorul cardului.
- Atacatorul are acces la card în condiții stabilite de el: poate să-l supună oricăror verificări și experimente dorește, pentru a avea acces la informațiile stocate.
  În particular, poate distruge cardul, pentru a vedea cum este construit.

Multe astfel de atacuri s-au materializat efectiv contra cardurilor de acces pay-TV și a cartelelor de telefon.

Atacuri ale proprietarului contra emițătorului de card

Sunt atacuri care au ca țintă integritatea și autenticitatea informațiilor stocate pe card.

Atacuri ale proprietarului contra emițătorului de card

Sunt atacuri care au ca țintă integritatea și autenticitatea informațiilor stocate pe card.

Multe din aceste informații sunt cunoscute de proprietarul de card; acesta dorește să le modifice fără acordul emițătorului de card (și – implicit – al fabricantului de card).

Atacuri ale proprietarului contra emițătorului de card

Sunt atacuri care au ca țintă integritatea și autenticitatea informațiilor stocate pe card.

Multe din aceste informații sunt cunoscute de proprietarul de card; acesta dorește să le modifice fără acordul emițătorului de card (și – implicit – al fabricantului de card).

Exemplu

La o cartelă de telefon prepaid, proprietarul (nu neapărat cel legal) poate încerca să modifice contul, cu scopul de a putea vorbi mai mult cu aceeași sumă.

Securitatea poate fi asigurată adăugând un mecanism de tip provocare/răspuns sau un lanț de funcții de dispersie inverse.

Atacuri ale proprietarului contra emițătorului de card

Sunt atacuri care au ca țintă integritatea și autenticitatea informațiilor stocate pe card.

Multe din aceste informații sunt cunoscute de proprietarul de card; acesta dorește să le modifice fără acordul emițătorului de card (și – implicit – al fabricantului de card).

Exemplu

La o cartelă de telefon prepaid, proprietarul (nu neapărat cel legal) poate încerca să modifice contul, cu scopul de a putea vorbi mai mult cu aceeași sumă.

Securitatea poate fi asigurată adăugând un mecanism de tip provocare/răspuns sau un lanț de funcții de dispersie inverse.

Toate precauțiile se bazează pe presupunerea că perimetrul de securitate al cardului este suficient de sigur.

Proprietarul de card contra programatorul de software

Precauțiile folosesc diverse protocoale preliminare de recunoaștere, bazate pe transformări neinversabile, pentru a avea siguranța că nu este posibil accesul la softul de pe card.
Se pleacă de la prezumția că nu există nici o legătură între proprietarul cardului și proprietarul softului.

Proprietarul de card contra programatorul de software

Precauțiile folosesc diverse protocoale preliminare de recunoaștere, bazate pe transformări neinversabile, pentru a avea siguranța că nu este posibil accesul la softul de pe card.
Se pleacă de la prezumția că nu există nici o legătură între proprietarul cardului și proprietarul softului.

Totuși, intrușii dovedesc o abilitate deosebită în construcția unor structuri hard adecvate lansării unor astfel de atacuri, structuri oferite adesea gratis pe Internet.

Proprietarul de terminale contra emițătorul de carduri

Într-un sistem închis la exterior (cum este cel al cartelelor de telefon prepaid), proprietarul de terminale este de asemenea și emițătorul de carduri (compania de telefoane îndeplinește ambele funcții).

Proprietarul de terminale contra emițătorul de carduri

Într-un sistem închis la exterior (cum este cel al cartelelor de telefon prepaid), proprietarul de terminale este de asemenea și emițătorul de carduri (compania de telefoane îndeplinește ambele funcții).
La sistemele deschise (când cele două entități sunt distincte), terminalul controlează toate comunicațiile între card și emițătorul de carduri.

Proprietarul de terminale contra emițătorul de carduri

Într-un sistem închis la exterior (cum este cel al cartelelor de telefon prepaid), proprietarul de terminale este de asemenea și emițătorul de carduri (compania de telefoane îndeplinește ambele funcții).
La sistemele deschise (când cele două entități sunt distincte), terminalul controlează toate comunicațiile între card și emițătorul de carduri.
Deci el poate totdeauna să falsifice mesajele care nu au legătură cu conținutul smartcardului; de exemplu să refuze înregistrarea tranzacției, să lanseze comenzi false etc.

De asemenea, terminalul poate omite intenționat completarea unor pași dintr-o tranzacție, facilitând astfel o fraudă sau creind dificultăți de management emițătorului de card.

De asemenea, terminalul poate omite intenționat completarea unor pași dintr-o tranzacție, facilitând astfel o fraudă sau creind dificultăți de management emițătorului de card.

Exemplu

Terminalul nu modifică suma de pe card (scăzând suma cheltuită).

De asemenea, terminalul poate omite intenționat completarea unor pași dintr-o tranzacție, facilitând astfel o fraudă sau creind dificultăți de management emițătorului de card.

Exemplu

Terminalul nu modifică suma de pe card (scăzând suma cheltuită). Terminalul completează o tranzacție dar nu oferă serviciul respectiv (de exemplu, scoate banii, dar nu oferă marfa).

De asemenea, terminalul poate omite intenționat completarea unor pași dintr-o tranzacție, facilitând astfel o fraudă sau creind dificultăți de management emițătorului de card.

Exemplu

Terminalul nu modifică suma de pe card (scăzând suma cheltuită). Terminalul completează o tranzacție dar nu oferă serviciul respectiv (de exemplu, scoate banii, dar nu oferă marfa).

De obicei atacurile sunt prevenite printr-o monitorizare permanentă la ambele capete a protocoalelor de comunicare.

Atacuri ale emițătorului contra proprietarului de card

Cele mai multe sisteme de smartcard se bazează pe supoziția că emițătorul de carduri slujește interesele proprietarilor de card.

Atacuri ale emițătorului contra proprietarului de card

Cele mai multe sisteme de smartcard se bazează pe supoziția că emițătorul de carduri slujește interesele proprietarilor de card. Majoritatea atacurilor încalcă conceptul de confidențialitate.

Exemplu

Astfel smartcardurile care efectuează plăți trebuie să asigure menținerea proprietăților de anonimitate și fără legături (unlinkability).

Atacuri ale emițătorului contra proprietarului de card

Cele mai multe sisteme de smartcard se bazează pe supoziția că emițătorul de carduri slujește interesele proprietarilor de card. Majoritatea atacurilor încalcă conceptul de confidențialitate.

Exemplu

Astfel smartcardurile care efectuează plăți trebuie să asigure menținerea proprietăților de anonimitate și fără legături (unlinkability).

Aceste atacuri (unele doar slăbiciuni ale sistemului smartcard) se pot efectua fără ca proprietarul de card să fie conștient de existența lor, sau fără să fie avertizat asupra lor.

Atacuri ale emițătorului contra proprietarului de card

Cele mai multe sisteme de smartcard se bazează pe supoziția că emițătorul de carduri slujește interesele proprietarilor de card. Majoritatea atacurilor încalcă conceptul de confidențialitate.

Exemplu

Astfel smartcardurile care efectuează plăți trebuie să asigure menținerea proprietăților de anonimitate și fără legături (unlinkability).

Aceste atacuri (unele doar slăbiciuni ale sistemului smartcard) se pot efectua fără ca proprietarul de card să fie conștient de existența lor, sau fără să fie avertizat asupra lor.

Majoritatea lor pot fi realizate de emițătorul de carduri doar în colaborare cu fabricantul de carduri și cu programatorul de software.

Fabricantul de carduri contra proprietarul de date

Anumite designuri de fabricare pot avea efecte negative destul de importante privind securitatea datelor de pe card.

Fabricantul de carduri contra proprietarul de date

Anumite designuri de fabricare pot avea efecte negative destul de importante privind securitatea datelor de pe card.

Fabricantul de carduri are numeroase posibilități de a-și asigura baza unor atacuri ulterioare asupra informațiilor de pe card.

Fabricantul de carduri contra proprietarul de date

Anumite designuri de fabricare pot avea efecte negative destul de importante privind securitatea datelor de pe card.

Fabricantul de carduri are numeroase posibilități de a-și asigura baza unor atacuri ulterioare asupra informațiilor de pe card.
Nici un fabricant de carduri nu a asigurat până acum un sistem de operare sigur, fără vulnerabilități (mai mult sau mai puțin evidente).

Fabricantul de carduri contra proprietarul de date

Anumite designuri de fabricare pot avea efecte negative destul de importante privind securitatea datelor de pe card.

Fabricantul de carduri are numeroase posibilități de a-și asigura baza unor atacuri ulterioare asupra informațiilor de pe card.
Nici un fabricant de carduri nu a asigurat până acum un sistem de operare sigur, fără vulnerabilități (mai mult sau mai puțin evidente).
În plus, prin implementarea diverselor protocoale, el poate aranja unele canale subliminale care să permită aflarea cheilor sau a altor date de pe card.

Este posibil ca pentru o aplicație pe smartcard să existe o altă aplicație care să ruleze simultan pe același smartcard.

S-a demonstrat că dacă avem un protocol sigur, se poate crea alt protocol – tot sigur – care să-l spargă pe primul, dacă ambele rulează pe același echipament și folosesc aceleași chei.

Este posibil ca pentru o aplicație pe smartcard să existe o altă aplicație care să ruleze simultan pe același smartcard.

S-a demonstrat că dacă avem un protocol sigur, se poate crea alt protocol – tot sigur – care să-l spargă pe primul, dacă ambele rulează pe același echipament și folosesc aceleași chei.

Un sistem de operare care permite mai multor utilizatori să ruleze programe pe același smartcard ridică numeroase probleme de securitate.

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

1. Cum poate ști utilizatorul (sau designerul de card) că programul rulează atunci când cardul este inserat într-un terminal ?

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

1. Cum poate ști utilizatorul (sau designerul de card) că programul rulează atunci când cardul este inserat într-un terminal ?
2. Cum poate fi el sigur că programul interacționează cu acel terminal și nu cu un alt program ?

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

1. Cum poate ști utilizatorul (sau designerul de card) că programul rulează atunci când cardul este inserat într-un terminal ?
2. Cum poate fi el sigur că programul interacționează cu acel terminal și nu cu un alt program ?
3. Cum poate un program, care ajunge la concluzia că este compromis, să termine în siguranță și să semnaleze utilizatorului că trebuie înlocuit ?

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

1. Cum poate ști utilizatorul (sau designerul de card) că programul rulează atunci când cardul este inserat într-un terminal ?
2. Cum poate fi el sigur că programul interacționează cu acel terminal și nu cu un alt program ?
3. Cum poate un program, care ajunge la concluzia că este compromis, să termine în siguranță și să semnaleze utilizatorului că trebuie înlocuit ?
4. Ce atacuri devin posibile atunci când un card anunță că nu mai lucrează deoarece nu mai este sigur ?

Dar, chiar și cu un sistem de operare (teoretic) sigur, securitatea interfeței cu utilizatorul constituie permanent un risc de securitate.

1. Cum poate ști utilizatorul (sau designerul de card) că programul rulează atunci când cardul este inserat într-un terminal ?
2. Cum poate fi el sigur că programul interacționează cu acel terminal și nu cu un alt program ?
3. Cum poate un program, care ajunge la concluzia că este compromis, să termine în siguranță și să semnaleze utilizatorului că trebuie înlocuit ?
4. Ce atacuri devin posibile atunci când un card anunță că nu mai lucrează deoarece nu mai este sigur ?

Mai puțin evidente sunt implementări intenționate de generatori slabi de numere pseudo-aleatoare, sau alte module criptografice de calitate inferioară privind securitatea.

Tehnici invazive

Definiție

Un atac asupra unui smartcard este "invaziv" dacă implică o modificare vizibilă a dispozitivului.

Tehnici invazive

Definiție

Un atac asupra unui smartcard este "invaziv" dacă implică o modificare vizibilă a dispozitivului.

De obicei tehnicile invazive presupun distrugerea totală a cipului smartcard-ului.

Tehnici invazive

Definiție

Un atac asupra unui smartcard este "invaziv" dacă implică o modificare vizibilă a dispozitivului.

De obicei tehnicile invazive presupun distrugerea totală a cipului smartcard-ului.
În plus, în timp ce atacurile non-invazive pot fi efectuate prin sustragerea pentru o scurtă perioadă de timp a unui dispozitiv smartcard, atacurile invazive pot necesita ore de muncă specializată în laboratoare, fiind accesibile doar atacatorilor foarte experimentați și cu suport financiar puternic.

Tehnici invazive

Definiție

Un atac asupra unui smartcard este "invaziv" dacă implică o modificare vizibilă a dispozitivului.

De obicei tehnicile invazive presupun distrugerea totală a cipului smartcard-ului.
În plus, în timp ce atacurile non-invazive pot fi efectuate prin sustragerea pentru o scurtă perioadă de timp a unui dispozitiv smartcard, atacurile invazive pot necesita ore de muncă specializată în laboratoare, fiind accesibile doar atacatorilor foarte experimentați și cu suport financiar puternic.
Există o probabilitate neglijabilă ca un atac invaziv să se realizeze fără cunoștința utilizatorului.

Toate atacurile invazive încep prin scoaterea cip-ului din suportul de plastic.

Toate atacurile invazive încep prin scoaterea cip-ului din suportul de plastic.

- Dacă cipul este proiectat să efectueze funcții speciale, există o anumită șansă ca cei ce l-au proiectat să nu fi ținut cont de principiul lui Kerckhoff.

  Astfel, componentele cipului pot fi analizate iar proiectul inițial poate fi descoperit ("reverse engineering").

Toate atacurile invazive încep prin scoaterea cip-ului din suportul de plastic.

- Dacă cipul este proiectat să efectueze funcții speciale, există o anumită șansă ca cei ce l-au proiectat să nu fi ținut cont de principiul lui Kerckhoff.

  Astfel, componentele cipului pot fi analizate iar proiectul inițial poate fi descoperit ("reverse engineering").
- *Microsondarea*: atac care presupune interacțiunea directă cu componentele cipului.

Toate atacurile invazive încep prin scoaterea cip-ului din suportul de plastic.

- Dacă cipul este proiectat să efectueze funcții speciale, există o anumită șansă ca cei ce l-au proiectat să nu fi ținut cont de principiul lui Kerckhoff.

  Astfel, componentele cipului pot fi analizate iar proiectul inițial poate fi descoperit ("reverse engineering").
- *Microsondarea*: atac care presupune interacțiunea directă cu componentele cipului.

  Se încalcă prezumția conform căreia smartcard-ul poate fi accesat numai prin intermediul unui cititor de carduri.

Toate atacurile invazive încep prin scoaterea cip-ului din suportul de plastic.

- Dacă cipul este proiectat să efectueze funcții speciale, există o anumită șansă ca cei ce l-au proiectat să nu fi ținut cont de principiul lui Kerckhoff.

  Astfel, componentele cipului pot fi analizate iar proiectul inițial poate fi descoperit ("reverse engineering").
- *Microsondarea*: atac care presupune interacțiunea directă cu componentele cipului.

  Se încalcă prezumția conform căreia smartcard-ul poate fi accesat numai prin intermediul unui cititor de carduri.

Exemplu

Cu două microsonde, orice bit din EEPROM poate fi setat sau resetat (aceasta face banală, de exemplu, extragerea unei chei).Similar, cu un microscop și un laser-cutter, orice bit din ROM poate fi modificat.

Atacul Biham Shamir

Un atac (construit de Biham și Shamir împotriva unei implementări hardware a *DES*) a folosit un laser-cutter pentru a distruge o poartă din implementarea hardware a unui bloc de criptare cunoscut, anume ultimul bit din registrul ce duce ieșirea unei runde în intrarea rundei următoare.

Atacul Biham Shamir

Un atac (construit de Biham și Shamir împotriva unei implementări hardware a *DES*) a folosit un laser-cutter pentru a distruge o poartă din implementarea hardware a unui bloc de criptare cunoscut, anume ultimul bit din registrul ce duce ieșirea unei runde în intrarea rundei următoare.
Blocarea bit-ului cel mai nesemnificativ are ca efect blocarea pe 0 a ultimului bit întors de funcția de rundă.
Prin compararea celor mai nesemnificativi 6 biți din jumătatea stângă și cea dreaptă, pot fi descoperiți câțiva biți din cheie.

Atacul Biham Shamir

Un atac (construit de Biham și Shamir împotriva unei implementări hardware a *DES*) a folosit un laser-cutter pentru a distruge o poartă din implementarea hardware a unui bloc de criptare cunoscut, anume ultimul bit din registrul ce duce ieșirea unei runde în intrarea rundei următoare.
Blocarea bit-ului cel mai nesemnificativ are ca efect blocarea pe 0 a ultimului bit întors de funcția de rundă.
Prin compararea celor mai nesemnificativi 6 biți din jumătatea stângă și cea dreaptă, pot fi descoperiți câțiva biți din cheie.
Atacul funcționează în multe aplicații la care cardul folosește tranzacții succesive pentru a raporta starea internă emitentului.

Atacul Biham Shamir

Un atac (construit de Biham și Shamir împotriva unei implementări hardware a *DES*) a folosit un laser-cutter pentru a distruge o poartă din implementarea hardware a unui bloc de criptare cunoscut, anume ultimul bit din registrul ce duce ieșirea unei runde în intrarea rundei următoare.
Blocarea bit-ului cel mai nesemnificativ are ca efect blocarea pe 0 a ultimului bit întors de funcția de rundă.
Prin compararea celor mai nesemnificativi 6 biți din jumătatea stângă și cea dreaptă, pot fi descoperiți câțiva biți din cheie.
Atacul funcționează în multe aplicații la care cardul folosește tranzacții succesive pentru a raporta starea internă emitentului.
Protecție: Se poate atașa sistemului o auto-testare care criptează și decriptează un text arbitrar folosind o cheie aleatoare, iar apoi compară rezultatul cu blocul original.

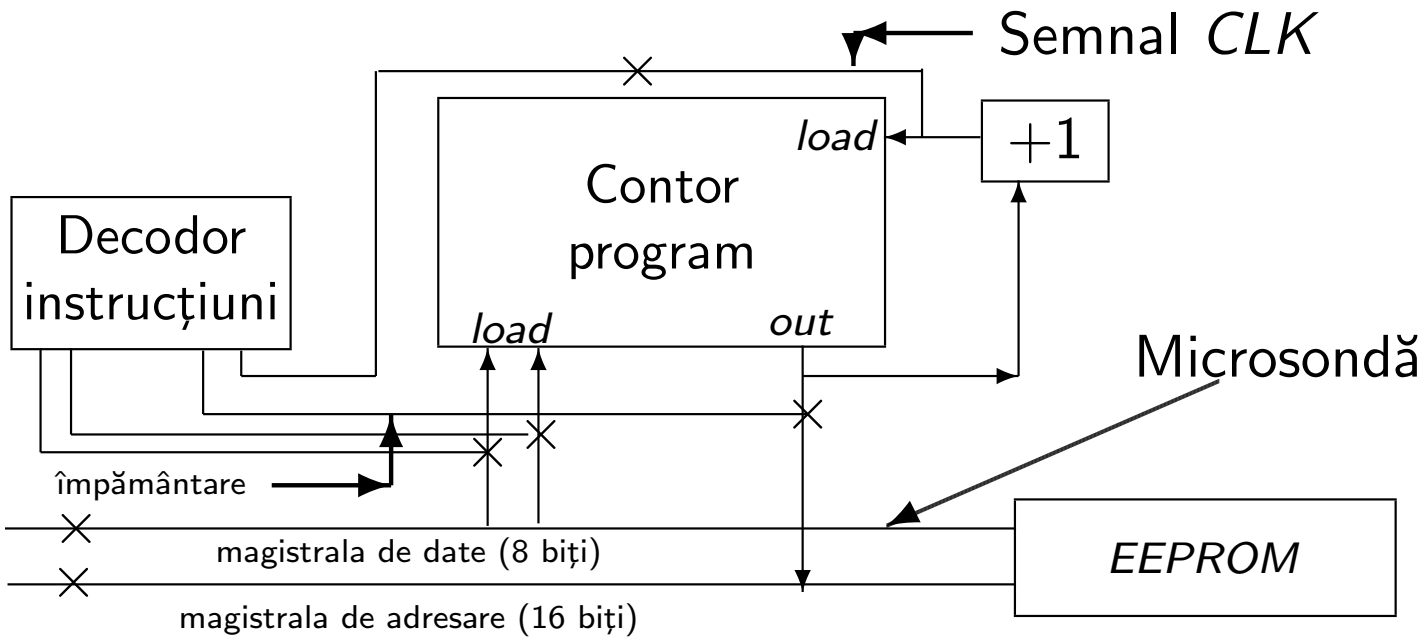
Un alt atac tipic presupune deconectarea *CPU* de magistrala de date, lăsând conectate numai *EEPROM*-ul și o componentă a *CPU* care să asigure accesul la citire.

Un alt atac tipic presupune deconectarea *CPU* de magistrala de date, lăsând conectate numai *EEPROM*-ul și o componentă a *CPU* care să asigure accesul la citire.
Pentru a citi toate celulele de memorie fără ajutorul software-ului card-ului trebuie folosită o componentă a *CPU*-ului drept contor de adrese pentru a accesa toate celulele de memorie.

Un alt atac tipic presupune deconectarea *CPU* de magistrala de date, lăsând conectate numai *EEPROM*-ul și o componentă a *CPU* care să asigure accesul la citire.
Pentru a citi toate celulele de memorie fără ajutorul software-ului card-ului trebuie folosită o componentă a *CPU*-ului drept contor de adrese pentru a accesa toate celulele de memorie.
Contorul de program este implicit incrementat automat în timpul fiecărui ciclu de instrucțiuni și folosit la citirea următoarei adrese, ceea ce îl face foarte potrivit pentru rolul de generator de secvențe de adrese.

Un alt atac tipic presupune deconectarea *CPU* de magistrala de date, lăsând conectate numai *EEPROM*-ul și o componentă a *CPU* care să asigure accesul la citire.
Pentru a citi toate celulele de memorie fără ajutorul software-ului card-ului trebuie folosită o componentă a *CPU*-ului drept contor de adrese pentru a accesa toate celulele de memorie.
Contorul de program este implicit incrementat automat în timpul fiecărui ciclu de instrucțiuni și folosit la citirea următoarei adrese, ceea ce îl face foarte potrivit pentru rolul de generator de secvențe de adrese.

Totuși, procesorul trebuie împiedicat să execute instrucțiuni cum ar fi *jump, call* sau *return*, care ar deturna contorul de program de la citirea secvențială.
Mici modificări ale decodorului de instrucțiuni sau ale circuitelor contorului de program pot avea acest efect.



Odată ce aceste măsuri au fost luate, *Oscar* are nevoie de un singur ac pentru microsondare, sau o sondă electro-optică pentru a citi întregul conținut al *EEPROM*-ului.

Se pot adăuga contoare auxiliare care să reseteze procesorul dacă nu se execută instrucțiuni *jump*, *call* sau *return* într-un anumit interval de timp.

Se pot adăuga contoare auxiliare care să reseteze procesorul dacă nu se execută instrucțiuni *jump, call* sau *return* într-un anumit interval de timp.
Totuși astfel de dispozitive pot fi dezactivate prin modificări minore ale circuitelor, așa că de obicei protecția este incorporată în structura cip-ului; de exemplu prin folosirea unui contor de program care să nu poată acoperi întregul spațiu de adrese.

Se pot adăuga contoare auxiliare care să reseteze procesorul dacă nu se execută instrucțiuni *jump, call* sau *return* într-un anumit interval de timp.
Totuși astfel de dispozitive pot fi dezactivate prin modificări minore ale circuitelor, așa că de obicei protecția este incorporată în structura cip-ului; de exemplu prin folosirea unui contor de program care să nu poată acoperi întregul spațiu de adrese.
Un contor de program pe 16 biți poate fi înlocuit ușor cu o combinație dintr-un contor O de deplasare pe 7 biți și un registru de segment R pe 16 biți, astfel încât adresa accesată să fie $S + O$.

Se pot adăuga contoare auxiliare care să reseteze procesorul dacă nu se execută instrucțiuni *jump, call* sau *return* într-un anumit interval de timp.
Totuși astfel de dispozitive pot fi dezactivate prin modificări minore ale circuitelor, așa că de obicei protecția este incorporată în structura cip-ului; de exemplu prin folosirea unui contor de program care să nu poată acoperi întregul spațiu de adrese.
Un contor de program pe 16 biți poate fi înlocuit ușor cu o combinație dintr-un contor O de deplasare pe 7 biți și un registru de segment R pe 16 biți, astfel încât adresa accesată să fie $S + O$. Contorul de deplasare resetează procesorul dacă ajunge la valoarea maximă. Procesorului îi va fi astfel imposibil să execute mai mult de 127 octeți de cod-mașină fără *jump*, și acest lucru nu poate fi schimbat de către un atacator prin mijloace simple.

Se pot adăuga contoare auxiliare care să reseteze procesorul dacă nu se execută instrucțiuni *jump, call* sau *return* într-un anumit interval de timp.
Totuși astfel de dispozitive pot fi dezactivate prin modificări minore ale circuitelor, așa că de obicei protecția este incorporată în structura cip-ului; de exemplu prin folosirea unui contor de program care să nu poată acoperi întregul spațiu de adrese.
Un contor de program pe 16 biți poate fi înlocuit ușor cu o combinație dintr-un contor O de deplasare pe 7 biți și un registru de segment R pe 16 biți, astfel încât adresa accesată să fie $S + O$. Contorul de deplasare resetează procesorul dacă ajunge la valoarea maximă. Procesorului îi va fi astfel imposibil să execute mai mult de 127 octeți de cod-mașină fără *jump*, și acest lucru nu poate fi schimbat de către un atacator prin mijloace simple.

Oscar poate încerca să crească numărul de iterații din cicluri software care citesc șiruri de octeți din memorie, pentru a avea acces la toți octeții.

Oscar poate încerca să crească numărul de iterații din cicluri software care citesc șiruri de octeți din memorie, pentru a avea acces la toți octeții.
Acest deziderat se poate obține folosind o microsondă care să producă o perturbație direct pe magistrala de date.
Programatorii care utilizează contoare pe 16 biți trebuie să țină cont de acest lucru.

Oscar poate încerca să crească numărul de iterații din cicluri software care citesc șiruri de octeți din memorie, pentru a avea acces la toți octeții.
Acest deziderat se poate obține folosind o microsondă care să producă o perturbație direct pe magistrala de date.
Programatorii care utilizează contoare pe 16 biți trebuie să țină cont de acest lucru.
Contracarare: Cele mai multe sisteme de operare pentru smartcard-uri criptează datele importante de pe *EEPROM*; astfel este dificil pentru un intrus să obțină text clar direct din *EEPROM*.

Oscar poate încerca să crească numărul de iterații din cicluri software care citesc șiruri de octeți din memorie, pentru a avea acces la toți octeții.
Acest deziderat se poate obține folosind o microsondă care să producă o perturbație direct pe magistrala de date.
Programatorii care utilizează contoare pe 16 biți trebuie să țină cont de acest lucru.
Contracarare: Cele mai multe sisteme de operare pentru smartcard-uri criptează datele importante de pe *EEPROM*; astfel este dificil pentru un intrus să obțină text clar direct din *EEPROM*.

Aceasta nu este de fapt o soluție, decât dacă criptarea depinde de un secret (cum ar fi *PIN*-ul introdus de utilizator).

Tehnici non-invazive de atac

Un atac non-invaziv asupra unui dispozitiv smartcard este oarecum mai limitat, dar – deoarece nu face nici o modificare asupra smartcardului – este foarte greu de descoperit.

Tehnici non-invazive de atac

Un atac non-invaziv asupra unui dispozitiv smartcard este oarecum mai limitat, dar – deoarece nu face nici o modificare asupra smartcardului – este foarte greu de descoperit.

Exemplu

Dacă s-ar găsi cheia privată a unui dispozitiv de semnătură fără ca Alice să observe, se pot falsifica documente în numele ei mult timp înainte de a se descoperi infracțiunea.

Tehnici non-invazive de atac

Un atac non-invaziv asupra unui dispozitiv smartcard este oarecum mai limitat, dar – deoarece nu face nici o modificare asupra smartcardului – este foarte greu de descoperit.

Exemplu

Dacă s-ar găsi cheia privată a unui dispozitiv de semnătură fără ca Alice să observe, se pot falsifica documente în numele ei mult timp înainte de a se descoperi infracțiunea.

Unele atacuri non-invazive pot decurge într-un timp scurt și folosind resurse hardware uzuale (eventual modificate puțin).

Tehnici non-invazive de atac

Un atac non-invaziv asupra unui dispozitiv smartcard este oarecum mai limitat, dar – deoarece nu face nici o modificare asupra smartcardului – este foarte greu de descoperit.

Exemplu

Dacă s-ar găsi cheia privată a unui dispozitiv de semnătură fără ca Alice să observe, se pot falsifica documente în numele ei mult timp înainte de a se descoperi infracțiunea.

Unele atacuri non-invazive pot decurge într-un timp scurt și folosind resurse hardware uzuale (eventual modificate puțin). Ele trebuie să aibă loc în timp ce cardul operează în condiții normale; orice manipulare va avea ca obiect condițiile de mediu sau octeții care intră și ies din smartcard.

Atacul prin cronometrare

Diferite secvențe de octeți sunt trimise card-ului pentru a fi semnate cu cheia privată.
Informații cum ar fi timpul necesar efectuării unei operații și numărul de 0 și 1 din șirul de intrare sunt folosite ulterior de *Oscar* pentru a obține cheia privată.

Atacul prin cronometrare

Diferite secvențe de octeți sunt trimise card-ului pentru a fi semnate cu cheia privată.
Informații cum ar fi timpul necesar efectuării unei operații și numărul de 0 și 1 din șirul de intrare sunt folosite ulterior de *Oscar* pentru a obține cheia privată.
Fiind un atac cu text clar ales, este necesar ca *Oscar* să cunoască *PIN*-ul card-ului sau să-l păcălească pe utilizator să semneze șirurile de biți pe care acesta i le furnizează.

Atacul prin cronometrare

Diferite secvențe de octeți sunt trimise card-ului pentru a fi semnate cu cheia privată.
Informații cum ar fi timpul necesar efectuării unei operații și numărul de 0 și 1 din șirul de intrare sunt folosite ulterior de *Oscar* pentru a obține cheia privată.
Fiind un atac cu text clar ales, este necesar ca *Oscar* să cunoască *PIN*-ul card-ului sau să-l păcălească pe utilizator să semneze șirurile de biți pe care acesta i le furnizează.

Există măsuri de apărare la nivel hard împotriva acestui tip de atac dar nu toți producătorii de smartcard-uri le implementează.

Atacuri prin analiza consumului de energie

Se bazează pe măsurarea fluctuațiilor în curentul consumat de dispozitiv.

Atacuri prin analiza consumului de energie

Se bazează pe măsurarea fluctuațiilor în curentul consumat de dispozitiv.
Diferite instrucțiuni provoacă nivele diferite de activitate în decodorul de instrucțiuni și unitățile de calcul aritmetic; ele pot fi adesea distinse clar, iar pe baza lor pot fi reconstituite părți din algoritmi.

Atacuri prin analiza consumului de energie

Se bazează pe măsurarea fluctuațiilor în curentul consumat de dispozitiv.
Diferite instrucțiuni provoacă nivele diferite de activitate în decodorul de instrucțiuni și unitățile de calcul aritmetic; ele pot fi adesea distinse clar, iar pe baza lor pot fi reconstituite părți din algoritmi.
Aceste tehnici intră in categoria monitorizării informației și constituie o amenințare reală datorită faptului că pe piață există un număr mare de produse vulnerabile.

Atacuri prin analiza consumului de energie

Se bazează pe măsurarea fluctuațiilor în curentul consumat de dispozitiv.
Diferite instrucțiuni provoacă nivele diferite de activitate în decodorul de instrucțiuni și unitățile de calcul aritmetic; ele pot fi adesea distinse clar, iar pe baza lor pot fi reconstituite părți din algoritmi.
Aceste tehnici intră in categoria monitorizării informației și constituie o amenințare reală datorită faptului că pe piață există un număr mare de produse vulnerabile.

Atacurile sunt ușor de implementat, pot fi automatizate, au un cost scăzut și sunt non-invazive.

Analiza simplă a consumului de curent (*SPA*)

Presupune observarea directă a consumului de energie al sistemului pentru obținerea de informații despre secvența de instrucțiuni executată.

Analiza simplă a consumului de curent (*SPA*)

Presupune observarea directă a consumului de energie al sistemului pentru obținerea de informații despre secvența de instrucțiuni executată.

Atacatorii cu acces la tranzacții multiple și care posedă informații despre mecanismele interne specifice arhitecturii cip-ului folosit, pot fi periculoși.

Analiza simplă a consumului de curent (*SPA*)

Presupune observarea directă a consumului de energie al sistemului pentru obținerea de informații despre secvența de instrucțiuni executată.

Atacatorii cu acces la tranzacții multiple și care posedă informații despre mecanismele interne specifice arhitecturii cip-ului folosit, pot fi periculoși.

Kocher arată cum poate fi spart sistemul de criptare *DES* în cazul unei implementări hardware slabe, dar și cum poate fi evitat acest atac dacă sunt îndeplinite câteva condiții, cum ar fi de exemplu nelăsarea biților din cheie să fie folosiți în alegerea dintre două ramuri ale unui *jump*.

Analiza diferențială a consumului (*DPA*)

Se bazează pe faptul că memorarea unui bit 1 într-un flip - flop necesită de obicei mai multă energie decât memorarea unui bit 0.

Analiza diferențială a consumului (*DPA*)

Se bazează pe faptul că memorarea unui bit 1 într-un flip - flop necesită de obicei mai multă energie decât memorarea unui bit 0. De asemenea, schimbarea de stare cauzează un consum sporit de energie.

Analiza diferențială a consumului (*DPA*)

Se bazează pe faptul că memorarea unui bit 1 într-un flip - flop necesită de obicei mai multă energie decât memorarea unui bit 0. De asemenea, schimbarea de stare cauzează un consum sporit de energie.
Diferite instrucțiuni produc nivele variate de activitate în decodorul de instrucțiuni și în unitățile aritmetice; adesea ele pot fi diferențiate clar, și pe baza lor pot fi reconstruite părți din algoritm.

Analiza diferențială a consumului (*DPA*)

Se bazează pe faptul că memorarea unui bit 1 într-un flip - flop necesită de obicei mai multă energie decât memorarea unui bit 0. De asemenea, schimbarea de stare cauzează un consum sporit de energie.
Diferite instrucțiuni produc nivele variate de activitate în decodorul de instrucțiuni și în unitățile aritmetice; adesea ele pot fi diferențiate clar, și pe baza lor pot fi reconstruite părți din algoritm.

Componentele procesorului își schimbă valorile temporare stocate în momente diferite relativ la intervalul de oscilație al ceasului și pot fi separate prin măsurători de înaltă frecvență.

Este posibil un atac “reverse engineering” pentru un protocol sau un algoritm necunoscut.

Este posibil un atac "reverse engineering" pentru un protocol sau un algoritm necunoscut.
În general, semnalele care se scurg în timpul unei operații asimetrice tind să fie mult mai puternice decât acelea provenite de la algoritmii simetrici – de exemplu, din cauza complexității computaționale relativ ridicate a operațiilor de multiplicare.

Este posibil un atac "reverse engineering" pentru un protocol sau un algoritm necunoscut.
În general, semnalele care se scurg în timpul unei operații asimetrice tind să fie mult mai puternice decât acelea provenite de la algoritmii simetrici – de exemplu, din cauza complexității computaționale relativ ridicate a operațiilor de multiplicare.
De aceea implementarea măsurilor de apărare împotriva *SPA* și *DPA* poate fi dificilă.

Este posibil un atac "reverse engineering" pentru un protocol sau un algoritm necunoscut.
În general, semnalele care se scurg în timpul unei operații asimetrice tind să fie mult mai puternice decât acelea provenite de la algoritmii simetrici – de exemplu, din cauza complexității computaționale relativ ridicate a operațiilor de multiplicare.
De aceea implementarea măsurilor de apărare împotriva *SPA* și *DPA* poate fi dificilă.
DPA poate fi folosită pentru a sparge implementări ale aproximativ tuturor algoritmilor de criptare.

Este posibil un atac "reverse engineering" pentru un protocol sau un algoritm necunoscut.
În general, semnalele care se scurg în timpul unei operații asimetrice tind să fie mult mai puternice decât acelea provenite de la algoritmii simetrici – de exemplu, din cauza complexității computaționale relativ ridicate a operațiilor de multiplicare.
De aceea implementarea măsurilor de apărare împotriva *SPA* și *DPA* poate fi dificilă.
DPA poate fi folosită pentru a sparge implementări ale aproximativ tuturor algoritmilor de criptare.

Exemplu

O cheie secretă Twofish de 128 *biți a fost recuperată dintr-un smartcard după ce s-au monitorizat* 100 *criptări independente.*
În general DPA descoperă $1 - 2$ *biți de informație per criptare.*

Contramăsuri

- Se poate reduce dimensiunea semnalului folosind căi de execuție constantă a codului sau adăugând porți suplimentare care să compenseze consumul.

Contramăsuri

- Se poate reduce dimensiunea semnalului folosind căi de execuție constantă a codului sau adăugând porți suplimentare care să compenseze consumul.

  Din nefericire semnalul nu se poate reduce la zero, iar un atacator cu un număr (teoretic) infinit de mostre va fi capabil să folosească cu succes un atac bazat pe *DPA*.

Contramăsuri

- Se poate reduce dimensiunea semnalului folosind căi de execuție constantă a codului sau adăugând porți suplimentare care să compenseze consumul.

  Din nefericire semnalul nu se poate reduce la zero, iar un atacator cu un număr (teoretic) infinit de mostre va fi capabil să folosească cu succes un atac bazat pe *DPA*.
- Folosirea procedurilor neliniare de actualizare a cheii.

Contramăsuri

- Se poate reduce dimensiunea semnalului folosind căi de execuție constantă a codului sau adăugând porți suplimentare care să compenseze consumul.
  Din nefericire semnalul nu se poate reduce la zero, iar un atacator cu un număr (teoretic) infinit de mostre va fi capabil să folosească cu succes un atac bazat pe *DPA*.
- Folosirea procedurilor neliniare de actualizare a cheii.
  Sau, pot fi folosite modificări ad-hoc în procesele de utilizare a exponenților și de modificare a modulelor în cadrul sistemelor cu cheie publică (pentru a împiedica atacatorii să adune date în timpul unui număr mare de operații).

Atacuri prin generarea de erori

Se exercită presiuni asupra procesorului pentru a-l forța să execute operații ilegale sau să dea rezultate greșite.

Atacuri prin generarea de erori

Se exercită presiuni asupra procesorului pentru a-l forța să execute operații ilegale sau să dea rezultate greșite.

Exemplu

Se poate modifica tensiunea de alimentare și semnalul ceasului. Subtensionarea și supratensionarea pot fi folosite pentru a dezactiva circuitele de protecție sau pentru a forța procesoarele să efectueze operații greșite.

Atacuri prin generarea de erori

Se exercită presiuni asupra procesorului pentru a-l forța să execute operații ilegale sau să dea rezultate greșite.

Exemplu

Se poate modifica tensiunea de alimentare și semnalul ceasului. Subtensionarea și supratensionarea pot fi folosite pentru a dezactiva circuitele de protecție sau pentru a forța procesoarele să efectueze operații greșite.

Exemplu

O fluctuație intensă a unor parametri fizici, exact în momentul în care se efectuează verificarea PIN-ului.

Salturile condiționate creează ferestre de vulnerabilitate în stadiile de procesare a multor aplicații de securitate, care permit ocolirea barierelor criptografice prin simpla împiedicare a execuției codului care verifică dacă o autentificare are succes sau nu.

Salturile condiționate creează ferestre de vulnerabilitate în stadiile de procesare a multor aplicații de securitate, care permit ocolirea barierelor criptografice prin simpla împiedicare a execuției codului care verifică dacă o autentificare are succes sau nu.

Contramăsurile hardware includ generatori interni independenți de ceas, sincronizați doar cu frecvența externă de referință.

Salturile condiționate creează ferestre de vulnerabilitate în stadiile de procesare a multor aplicații de securitate, care permit ocolirea barierelor criptografice prin simpla împiedicare a execuției codului care verifică dacă o autentificare are succes sau nu.

Contramăsurile hardware includ generatori interni independenți de ceas, sincronizați doar cu frecvența externă de referință.
O abordare mai radicală – dar potențial mai benefică – este să se elimine complet ceasul, transformând procesoarele de card în circuite autocronometrate asincrone.

Salturile condiționate creează ferestre de vulnerabilitate în stadiile de procesare a multor aplicații de securitate, care permit ocolirea barierelor criptografice prin simpla împiedicare a execuției codului care verifică dacă o autentificare are succes sau nu.

Contramăsurile hardware includ generatori interni independenți de ceas, sincronizați doar cu frecvența externă de referință.
O abordare mai radicală – dar potențial mai benefică – este să se elimine complet ceasul, transformând procesoarele de card în circuite autocronometrate asincrone.
Ceasul extern ar fi folosit doar ca referință pentru comunicare și – prin urmare – perturbările de ceas ar cauza doar o corupere a datelor.

Sfârșit