Automated Fare Collection

- structure and operation of AFC systems
- attacking AFC systems
- security requirements
- proximity cards (ISO 14443)
- Mifare
- Calypso

Automated Fare Collection

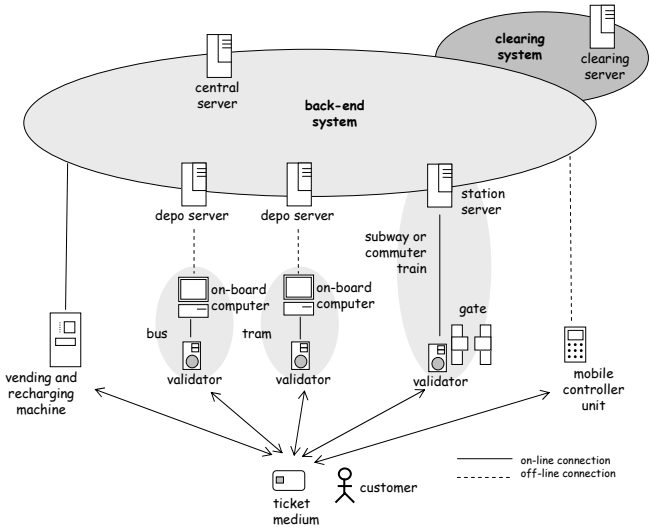
main idea:

replace public transportation tickets printed on paper with electronic tickets

- advantages:
  - easy collection of usage data by automatically logging transactions
  - better planning and optimization of services
  - flexible pricing schemes (e.g., based on distance or time)
  - easy access to combined services (e.g., park + ride, commuter train + city bus, etc.)
  - customer convenience
  - more difficult to counterfeit (especially smart cards)
- disadvantages:
  - price ?
  - location privacy ?

note: electronic ticketing doesn't prevent bilking → controllers are still needed

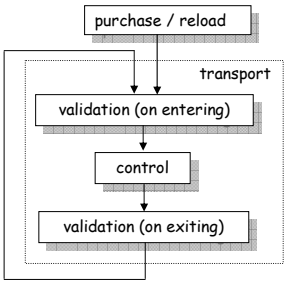
General structure of AFC systems



Ticket media

- magnetic strip cards
- contactless smart cards
  - memory cards
  - microprocessor cards
  - programmable microprocessor cards
- dual interface smart cards
  - multi-application cards (e.g. including an e-purse)
- virtual tickets (perhaps in the future)
  - mobile phone
  - PDA
  - MP3 player
  - wrist-watch
  - ...

Ticket life-cycle



- purchase / reload
  - transport contract may be logically associated with a counter
  - the counter needs to be reloaded from time to time
- validation (on entering)
  - the customer needs to validate its card on entering a vehicle or the subway system
    - stored contracts are checked
    - some counters may be decreased
    - transaction is logged on the card (and on the validator)
- control
  - in open transport systems (e.g., bus, tram), control is still needed (occasionally)
  - controller checks contracts and logged transactions on cards with a handheld mobile unit
- validation (on exiting)
  - in some cases, validation on exit is also needed (e.g., distance based charging)
  - encouraged by over-charging on entering

The back-end system

- depo servers and station servers
  - collect transaction logs
    - via pre-installed wired connections from static validators (e.g., gates)
    - via a dynamic wireless connection from vehicle on-board units when they return to the garage at the end of the day
  - forward transaction logs to the central server (via a virtual private network)
  - download blacklists and whitelists into validators and on-board units
- central server
  - collects transaction logs from the entire system
  - analyzes logs
    - computes usage data → input to the clearing system
    - computes statistics → input to service optimization
    - identifies anomalies → fraud detection
  - creates blacklists and whitelists and sends them to the depo and station servers
  - interacts with the clearing system
- clearing system
  - supports settlement between multiple transport operators

Attacking AFC systems – Who?

- average customer
  - small amount of resources and expertise
  - however, there are thousands of them
  - has access to own ticket media, may have access to other AFC equipments occasionally (e.g., public equipments or theft)
- expert outsider
  - small amount of resources, but considerable expertise
  - there are only a few of them
  - has access to a moderate number of tickets, may have access to other AFC equipments occasionally, may have access to special analysis tools
- misbehaving insider
  - small amount of resources
  - level of expertise may vary on a wide scale (simple user → system engineer)
  - very few of them
  - has access to a large amount of (blank) tickets and various AFC equipments on a regular basis
- criminal organization
  - huge amount of resources
  - can buy expertise and access to tickets and AFC equipments
- any combination of the above !!!

Attacking AFC systems – Why?

- service theft
  - using public transport services without paying for them
- large scale forgery
  - counterfeiting ticket media (cards) and/or transport contracts
  - setting up an illegal reload service and establishing a black market
- personal satisfaction
  - demonstrate that the system can be broken
- denial of service
  - cause loss to the PTO by vandalism, sabotage, etc.
- collecting personal data of customers
  - location information
  - special attributes (e.g., reasons for discounts)
  - usage habits

Attacking AFC systems – What and How?

- ticketing media
  - physical attacks
    - tampering with the ticketing medium
    - direct modification, deletion, or insertion of data
    - direct access to secret keys
  - side channel attacks
    - inferring secret data by observing and analyzing operational characteristics (e.g., timings)
- wireless communications (card – reader, vehicle – depo server)
  - eavesdropping, spoofing, replay, and jamming
- back-end system
  - breaking into the back-end system by "standard" hacking methods
- cryptographic protocols and algorithms
  - cryptanalytic attacks
  - using smart cards or AFC equipments as oracles
- non-technical attack methods
  - social engineering
  - criminal activities

Security requirements - preliminaries

- ticket medium type
  - passive ticket medium
    - can store data but cannot perform computation
  - active ticket medium
    - can control access to the stored data by performing (cryptographic) computations
- terminal type
  - on-line terminal
    - can communicate with the back-end servers during operation (e.g., validators at subway gates)
  - off-line terminal
    - can communicate with back-end servers only occasionally (e.g., validators on buses and trams)

Requirements (passive medium, off-line terminals)

- prevent illegal generation and manipulation of data stored on cards
  - data should be protected with a cryptographic MAC
  - terminal reads data with MAC, verifies MAC, and refuses the card if the verification is not successful
  - if data must be modified on the card (e.g., decrease of counter value), then the terminal computes a new MAC and writes the modified data and the MAC back to the card
  - terminal needs keys, terminal is off-line → keys must be stored on the terminal in a secure way
  - keys are usually stored in tamper resistant modules (SAM – security application module)
- prevent reading from one card and writing on another one
  - terminal should swallow the card
  - contactless cards ???
- prevent cloning of cards
  - using MAC doesn't prevent this
  - transactions must be logged, uploaded to the central server regularly, and analyzed → cloned cards can be identified and blacklisted
  - blacklisted cards are refused by terminals
  - clones can be used only for a limited amount of time

Requirements (passive medium, on-line terminals)

- enough to store an ID on the card
  - terminal reads ID and looks up corresponding data (e.g., transport contract) in the back-end systems in real-time
  - cloning cards doesn't generate value (counters are managed by the back-end system)
  - anomalies can be detected faster
  - cloned cards can be blacklisted immediately
  - blacklist is always available and up-to-date (back-end system itself uses the blacklist)
  - blacklist management is easier, blacklist size can be larger (no need to download into terminals regularly)
- could be quite secure, but not really used in practice
  - terminals are usually off-line, even if they could be on-line
  - the same card should work with off-line terminals too
- potential problem:
  - suppose that an attacker uses someone's ID
  - how the legitimate owner of the ID can detect and prove this?
  - if PTO doesn't accept customer claims, then customers become frustrated → PTO's image can be destroyed, and customers refuse to use the system
  - if PTO accepts customers' claims, then sharing IDs can become widespread
  - frequently reclaiming users are suspicious, but proving that they are dishonest may be very difficult

Requirements (active medium)

- active media are much more secure than passive ones
  - they can control access to stored data
  - they can run cryptographic protocols to authenticate terminals
- requirements are similar to those for secure authentication and key establishment protocols
  - prevent replay attacks
  - protocols should be designed in such a way that cards cannot be used as oracles to break secret keys
  - ensure atomicity of transactions (challenging in case of contactless cards)

Security requirements of the back-end system

- should have a detailed security policy
- should implement appropriate means to enforce the security policy
- a good starting point is BS 7799 or something similar
- in particular (since attackers can be insiders):
  - there should be a well identified responsible person for everything
  - critical operations should not be executable by a single entity
  - operations must be logged, log files must be protected and analyzed

Security mechanisms in AFC systems

- security protocols (entity authentication, transaction integrity protection, transaction atomicity, …)
- key diversification
  - how off-line terminals can handle thousands of cards?
  - each card has its own key
  - card key is generated from the card ID and a master key using a one-way function (compromise of the card key doesn't affect the master key)
  - terminals store only a few master keys, and compute card keys on-the-fly when they are needed
- logging, log analysis, fraud detection
- blacklists and whitelists
- physical protection (tamper resistant modules, non-modifiable IDs, …)
- standard security mechanisms for IT systems
  - firewalls, IDS, anti-virus software, VPN, …

ISO 14443

- specifications for contactless smart cards (proximity cards, ~10cm)
- two types:
  - ISO 14443 A → Mifare
  - ISO 14443 B → Calypso
- 4 parts:
  - 14443-1: physical characteristics
    - size, resistance to folding, UV and electromagnetic radiation, etc.
  - 14443-2: RF interface
    - modulation, bit encoding, synchronization, speed

	Olvasó → Kártya „A"	Olvasó → Kártya „B"	Kártya → Olvasó "A"	Kártya → Olvasó „B"
Moduláció	ASK 100%	ASK 10%	847 kHz ASK modulált alvivő (BPSK)	847 kHz ASK modulált alvivő (BPSK)
Bit kódolás	módosított Miller	NRZ (non return to zero)	Manchester kódolás	NRZ (non return to zero)
Szinkronizáció	bitorientált (start of frame, end of frame)	1 start és stop bit bájtonként	bitorientált (start of frame, end of frame)	1 start és stop bit bájtonként
Jelsebesség	106 kBd	106 kBd	106 kBd	106kBd

  - 14443-3: initialization and collision avoidance
  - 14443-4: data transfer protocol

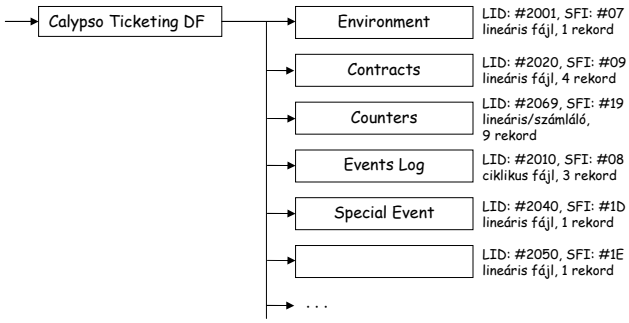
Mifare

- Mifare Ultralight
  - identification: unique serial number
  - access control:
    - 32 bit user programmable OTP area
    - field programmable read-only locking function
- Mifare Classic 1K and 4K
  - 1K: 16 sector, 4 block/sector, 16 byte/block
  - 4K: 8 sector with 16 block + 32 sector with 4 block
  - identification: unique serial number
  - access control: two keys per sector
  - authentication: three-pass authentication (ISO 9798-2) (Crypto1)
  - confidentiality and integrity: data encryption (Crypto1) on RF channel with replay protection
- Mifare DESfire
  - EEPROM: 4K, ISO 7816 file structure
  - identification: unique 7 byte serial number
  - access control: max 14 keys per application, access control on file level
  - authentication: three-pass authentication (ISO 9798-2) (3DES)
  - confidentiality and integrity: data encryption (DES/3DES) on RF channel with 4 byte MAC
- Mifare ProX
  - dual interface, 64K, DES and RSA

Calypso specifications

- developed in France in the mid 90's
- scope is not limited to the ticket media, but specifies upper layers too (e.g., key and SAM management)
- based on the following standards:
  - contactless interface: ISO 14443 B
  - file structure on card: ISO 7816-4
  - data structures (within files): ENV 1545

Calypso file structure on cards

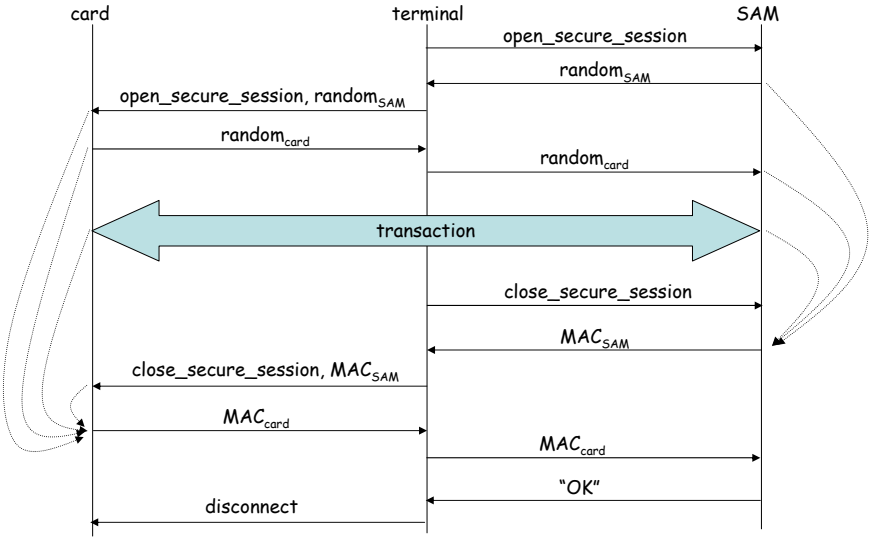


- environment:
  - information on transport application (version, transport network ID, validity)
  - card owner information (e.g., discounts)
- contracts
  - transport contracts (type, validity, limitations, vending information)
  - transport contracts may be logically associated with counters
- counters:
  - store value
  - operations: read, write, inc, dec
- events log:
  - stores last three transactions (type, time, place, transport contract number, charged counter value, …)
- special event:
  - e.g., card is refused by a validator

Calypso validation process

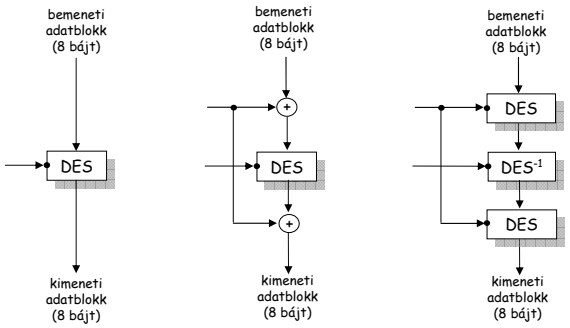
1. validator selects the Calypso applications on the card
2. establishment of a secure session between the card and the validator (SAM)
3. validator reads transport contract and associated counter
4. validator checks the conditions recorded in the transport contract, and checks the value of the associated counter
   - if transport contract is not valid, then reads the next contract
   - if counter value is too low, card is rejected
5. counter is decreased (decrease)
6. transaction is logged on the card (append record)
7. finishing the secure session with computing and sending MACs on the whole transaction (similar to SSL)
   - if the card doesn't receive the MAC from the validator, then the transaction is undone

Calypso "secure session" concept



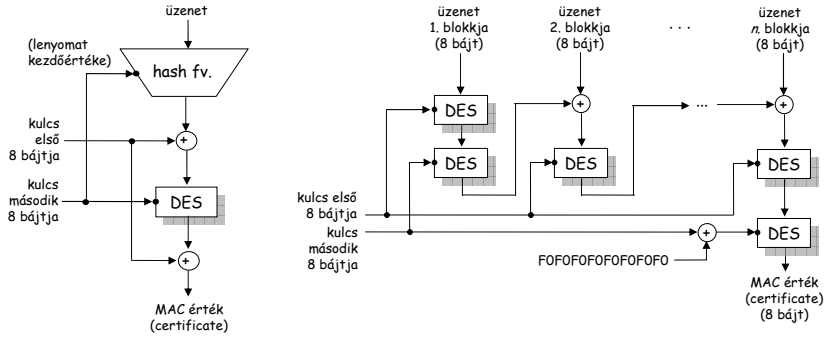
Cryptographic algorithms

- supported encryption algorithms:
  - DES (56 bit)
  - DESX (120 bit)
  - 3DES (112 bit)



Cryptographic algorithms

- supported MAC algorithms:
  - DESX MAC (112 bit)
  - 3DES CBC MAC (ISO 9797-1 / 4) (112 bit)



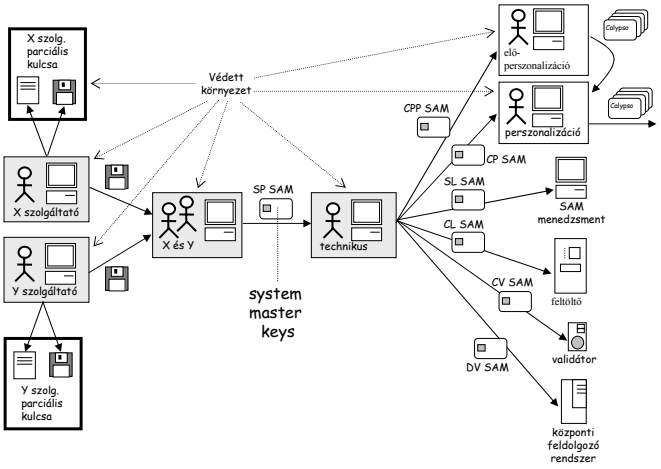
Calypso SAMs

- different equipment use different SAMs:
  - CPP-SAM: card pre-personalization equipments' SAM
  - CP-SAM: card personalization equipments' SAM
  - CL-SAM: card loading equipments' SAM
  - CV-SAM: card validator equipments' SAM
  - ...
- SAMs are smart cards too, they have a similar life-cycle to cards:
  - need to be personalized
  - can be written and read by authorized entities only
- equipments dealing with SAMs have their own SAM
  - SPP-SAM: SAM pre-personalization equipment's SAM
  - SP-SAM: SAM personalization equipment's SAM
  - SL-SAM: C*-SAM management (read, write, ...) equipment's SAM

Calypso keys

- card keys
  - cards have diversified keys
  - keys are loaded on the card during pre-personalization in a secure environment
  - different keys belong to different functions:
    - personalization key
    - reloading key
    - validation key
    - other keys for management of the card (read, write, …)
- SAM keys
  - SAMs have working keys and management keys
    - working keys are master keys from which card keys are generated on-the-file using the card ID
    - management keys are diversified keys that are used for the management of the SAM
  - SAMs store only those keys that are related to their function
    - e.g., a CV-SAM stores only the validation master key

Calypso key generation and distribution process



Summary of main security features in Calypso

- system master secret is stored in shares by participating PTOs
- shares of the master secret are pulled together only for the generation of the system master keys in a secure environment and under strict control (everything is logged and signed)
- SAM management has strict rules
- SP-SAMs are stored in a secure environment (vault, guards, surveillance, etc.)
- SAM personalization is done in the presence of at least two persons in a secure environment
- different master keys belong to different functions
- SAMs store only those master keys that are related to their function
- cards store only diversified keys
- card pre-personalization is done in a secure environment