

Simple but longterm card-terminal authorization protocol based on one time passwords - sketch of protocol

Prerequisites

- Each card has a unique id and stores its current state (ST), which is simultaneously a symmetric key used for secure communication with the terminal.
- Terminal stores a mapping from card ids to their current states. We assume that for a given card, the initial

Definitions

- $Card_{ID}$ - card IDs space $\{0, 1\}^{32}$
- R - challenges space $\{0, 1\}^{64}$
- K - key space $\{0, 1\}^{256}$
- Enc - encryption (AES)
- Dec - decryption (AES)
- $ACRT$ - acceptable card response time (exact value to be defined)
- $time()$ - function that returns current timestamp
- $f : Card_{ID} \rightarrow K$ - mapping from card ids to their current states

Authentication protocol (simple pre-shared key challenge-response authentication):

Terminal	Transmission	Card
1. Has f		Has ST
	$\leftarrow Card_{ID}$	
2. Take $r \in R$ uniformly at random. Let $t := time()$		
	$\rightarrow R$	
3.		$m_1 := Enc_{ST}(r)$
	$\leftarrow m_1$	
4. Let $t' := time()$. Check if $t' - t < ACRT$ (If not, abort.)		
Let $k := f(Card_{ID})$ and check if $Dec_K(m_1) = r$ (If not, abort.)		
Take $k' \in K \setminus \{k\}$ uniformly at random and update f so that $f(Card_{ID}) = k'$.		
$m_2 := Enc_k(k')$		
	$\rightarrow m_2$	
5.		$ST := Dec_{ST}(m_2)$

If the protocol is executed successfully, terminal opens the door to the secure location.

ASN.1 Documentation

```
CardProtocol DEFINITIONS ::= BEGIN
    CardHello ::= SEQUENCE {
cardId BIT STRING
    }

    RandomChallenge ::= SEQUENCE challenge BIT STRING
    StageOne ::= SEQUENCE oldState BIT STRING
    StageTwo ::= SEQUENCE newState BIT STRING
END
```