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

Prerequisites

- Each card has a unique ID ($Card_{ID}$) 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 state of the card and the corresponding state terminal holds are the same.

Definitions

- $\mathcal{ID}$ - card IDs space $\{0,1\}^{32}$
- $\mathcal{R}$ - challenges space $\{0,1\}^{64}$
- $\mathcal{K}$ - 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
- $f : \mathcal{ID} \to \mathcal{K}$ - mapping from card IDs to their current states

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

Terminal (f)	Transmission	Card ($Card_{ID}$, ST)
1.		
	$\leftarrow Card_{ID}$	
2. Take $r \in \mathcal{R}$ 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 $k' \in \mathcal{K} \setminus \{k\}$ 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
```