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 pair of their current and previous state. We assume that for a given card the initial state of the card and the corresponding current state that terminal holds are the same (initial previous state of the card stored by the terminal is taken at random).

Definitions

- $\mathcal{ID}$ - card IDs space ($\{0,1\}^{16}$)
- $\mathcal{R}$ - challenges space ($\{0,1\}^{64}$)
- $\mathcal{K}$ - key space ($\{0,1\}^{128} \times \{0,1\}^{256}$)
- Enc - encryption (AES in CFB mode with 256b key)
- Dec - decryption (AES in CFB mode with 256b key)
- ACRT - acceptable card response time (exact value to be defined)
- $time()$ - function that returns current time
- $f : \mathcal{ID} \to \mathcal{K} \times \mathcal{K}$ - mapping from card IDs to pair of states (previous and the current one).

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, show error message about card response being too long and abort.)		
Let $(k_{prev}, k_{curr}) := f(Card_{ID})$ and check if (1) $Dec_{k_{prev}}(m_1) = r$ or (2) $Dec_{k_{curr}}(m_1) = r$ (If (1) is fulfilled, show warning that terminal and card got desynchronized. If none is fulfilled, show error message about card being in an incorrect state, suggesting that it may have been cloned and abort).		
Let $k_{good} \in \{k_{prev}, k_{curr}\}$ be the one that fulfilled one of the equalities. Take $k' \in \mathcal{K} \backslash \{k_{good}\}$ uniformly at random and update f so that $f(Card_{ID}) = (k_{good}, k')$.		
$m_2 := Enc_{k_{good}}(k')$		
	$\rightarrow m_2$	
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
```