finite automata [7, 14]. One of the most famous longstanding open problems in automata theory, known as Černý Conjecture, states that for a given synchronizing DFA with n states one can always find a synchronizing word of length at most $(n-1)^2$. This conjecture was proven for numerous classes of automata, but the problem is still not solved in general case. The concept of synchronization has been also considered in coding theory [3, 8], parts orienting in manufacturing [5, 15], testing of reactive systems [19] and Markov Decision Processes [11, 12].

Allowing no outgoing transitions from some states for certain letters helps us to model a system for which certain actions cannot be accomplished while being in a specified state. This leads to the problem of finding a synchronizing word for a finite automaton, where transition function is not defined for all states. Notice that this is the most frequent case, if we use automata to model real-world systems. In practice, it rarely happens that a real system can be modeled with a DFA where transition function is total. The transition function is usually a partial one. This fact motivated many researchers to investigate the properties of partial finite automata relevant to practical problems of synchronization.

We know that, in general case, checking if a partial automaton can be synchronized is PSPACE-complete [13] even for binary alphabet [26] and those facts are essential in our latter considerations.

In this paper we present a public key cryptosystem utilizing fact, that checking if the PFA is carefully synchronizing is PSPACE-complete. This is however not the first attempt of trying to develop asymmetric cryptosystems with the notion of finite automata. Public key cryptography on finite automata with output is discussed in [21] and uses the notion of invertible automata to provide the hard computational problem, inevitable to design such cryptosystem.

The paper is organized as follows. In the section 2 we provide with the basic notions and facts about synchronization of automata. In the sections 3 and 4 we present basic method of encryption and decryption using our cryptosystem. In the section 5 we state couple of additional improvements to ensure better security. Finally we conclude the paper in the section 6 along with possible further research to the topic.

2 Preliminaries

Partial finite automaton (PFA) is an ordered tuple $\mathcal{A} = (Q, \Sigma, \delta)$, where Σ is a finite set of letters, Q is a finite set of states and $\delta : Q \times \Sigma^* \to Q$ is a transition function, possibly not everywhere defined. In this definition we omit initial and final states, since they are not relevant to the problem of synchronization. For $w \in \Sigma^*$ and $q \in Q$ we define $\delta(q, w)$ inductively: $\delta(q, \epsilon) = q$ and $\delta(q, aw) = \delta(\delta(q, a), w)$ for $a \in \Sigma$, where ϵ is the empty word and $\delta(q, a)$ is defined. A word $w \in \Sigma^*$ is called *carefully synchronizing* if there exists $\overline{q} \in Q$ such that for every $q \in Q$, $\delta(q, w) = \overline{q}$ and all transitions $\delta(q, w')$, where w' is any prefix of w, are defined. A PFA is called *carefully synchronizing* if it