we only can add b transitions to states that are some specified states of the copy of the public key. It is possible also to add the extension, that allows us to define the a-clusters for which we can define b transition outside of that specific sets B_i but to whatever state we want, even the other added a-cluster. However this extension would cause that in Lemma 5 it would be only $\{q_l^1, .., q_l^{|u|+1}\} \in P.w$ so the number of states added in such extensions would have been bounded by $\min(|Q_1'|, .., |Q_l'|)$ and also demanded modifications in Lemma 6 so we omitted that extension.
Observe also that point 2 of encryption procedure can be modified in many ways. For example one can choose to define more than one transition between copies of automata and in decryption section choose the one that has odd or even number in a ciphertext. We end up with several questions and open problems:

Question 1. *What is the most reasonable way to define lacking transitions in point 3?*

It is straightforward that if all lacking transitions in a copy of a public key are defined within the same copy, it would result with $|u| + 1$ connected automata which are not connected between each other, and that simplifies the attack on the cryptosystem.

Question 2. *What is the most reasonable way to define transitions in point 4?*

We have defined step 4 in an abstract way, so to investigate many versions of adding those "obfuscating" $\{0, 1\}$ transitions.

Question 3. *Find an algorithm that generates pairs of public and private keys.*

We believe that the most promising approach will be to construct a PFA that is carefully synchronized by a given w. We also want to investigate if it is possible to design an algorithm that for a given word w generates n non-isomorphic PFA's that are carefully synchronized by w. Having that one could take as a public key a tuple of n automata that are synchronized by the same word w. In that case, all methods presented in the paper would need only slight modifications to work properly.

References

[1] M. Berlinkov. On Two Algorithmic Problems about Synchronizing Automata. *Language and Automata Theory and Applications*, pages 61–67, 2014.

[2] M. Berlinkov and M. Szykuła. Algebraic Synchronization Criterion and Computing Reset Words. *Information Sciences*, 369:718–730, 2016.

[3] M. T. Biskup and W. Plandowski. Shortest Synchronizing Strings for Huffman Codes. *Theoretical Computer Science*, 410:3925–3941, 2009.

[4] J. H. Ellis. The Possibility of Non-Secret Digital Encryption. 1970.