INTRODUCTION

THE problem of private searching on streaming data was first introduced by Ostrovsky and Skeith. It was motivated by one of the tasks of the intelligence community, that is, how to collect potentially useful information from huge volumes of streaming data flowing through a public ever. However, that data which is potentially useful and raises a red flag is often classified and satisfies secret search criteria. The challenge is thus how to keep the search criteria classified even if the program residing in the public server falls into adversary’s hands. This problem has many applications for the purpose of intelligence gathering. For example, in airports one can use this technique to find if any of hundreds of passenger lists has a name from a possible list of terrorists and, if so, to find his/hers itinerary without revealing the secret terrorists list. The first solution for private searching on streaming data was given by Ostrovsky and Skeith. It was built on the concept of public-key program obfuscation, where an obfuscator compiles a given program f from a complexity class C into a pair of algorithms (F; Dec), such that DecðFðxÞÞ ¼ fðxÞ for any input x and it is impossible to distinguish for any polynomial time adversary which f from C was used to produce a given code for F. The basic idea can

be briefly described as follows. Assume that the public dictionary of potential keywords is D ¼ fw1; w2; . . . ; wjDjg. To search for documents containing one or more of classified keywords K ¼ fk1; k2; . . . ; kjKjg \_ D, the client generates a public/private key pair of a public key cryptosystem and constructs a program F, composed of an encrypted dictionary EðDÞ from K and a buffer IB which will store matching documents. Then the client dispatches the program F to a public server, where F filters a streaming documents and stores the encryptions of matching documents in the buffer IB. After the buffer IB returns, the client decrypts the buffer and retrieves the matching documents. Because both the keywords and the buffer are encrypted, the search criteria are kept classified to the public.

On the basis of this idea, several solutions for private searching on streaming data have been proposed in literature as follows: 1. Ostrovsky and Skeith gave two solutions for private searching on streaming data. One is based on the Paillier cryptosystem and allows to search for documents satisfying a disjunctive condition k1 \_ k2 \_ \_ \_ \_ \_ kjKj, i.e., containing one or more

classified keywords. Another is based on the Boneh et al. cryptosystem and can search for documents satisfying an AND of two sets of keywords. 2. Bethencourt et al. also gave a solution to search for documents satisfying a condition k1 \_ k2 \_ \_ \_ \_ \_ kjKj. Like the idea of, an encrypted dictionary is used. However, rather than using one large buffer and attempting to avoid collisions like Bethencourt et al. stored the matching documents in three buffers and retrieved them by solving linear systems.