Aggregating CL-Signatures Revisited: Extended Functionality and Better Efficiency

Kwangsu Lee¹,²,*, Dong Hoon Lee¹,**, and Moti Yung²,³

¹ CIST, Korea University, Korea
{guspin,donghlee}@korea.ac.kr
² Columbia University, USA
{kwangsu,moti}@cs.columbia.edu
³ Google Inc., USA

Abstract. Aggregate signature is public-key signature that allows anyone to aggregate different signatures generated by different signers on different messages into a short (called aggregate) signature. The notion has many applications where compressing the signature space is important: in infrastructure: secure routing protocols, in security: compressed certificate chain signature, in signing incrementally changed data: such as software module authentications, and in transaction systems: like in secure high-scale repositories and logs, typical in financial transactions. In spite of its importance, the state of the art of the primitive is such that it has not been easy to devise a suitable aggregate signature scheme that satisfies the conditions of real applications, with reasonable parameters: short public key size, short aggregate signatures size, and efficient aggregate signing/verification. In this paper, we propose two aggregate signature schemes based on the Camenisch-Lysyanskaya (CL) signature scheme whose security is reduced to that of CL signature (i.e., secure under the LRSW assumption) which substantially improve efficiency conditions for real applications. The first scheme is an “efficient sequential aggregate signature” scheme with the shortest size public key, to date, and very efficient aggregate verification. The second scheme is an “efficient synchronized aggregate signature” scheme with a very short public key size, and with the shortest (to date) size of aggregate signatures among synchronized aggregate signature schemes. Signing and aggregate verification are very efficient. Furthermore, our schemes are compatible: a signer of our aggregate signature schemes can dynamically use two modes of aggregation “sequential” and “synchronized,” employing the same private/public key.

Keywords: Public-key signature, Aggregate information applications, Aggregate signature, CL signature, Bilinear map.

* Supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (NIPA-2013-H0301-13-3007) supervised by the NIPA (National IT Industry Promotion Agency).

** Supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0029121).
1 Introduction

Public-key signature (PKS) is a central cryptographic primitive with numerous applications. However, constructing a PKS scheme that is efficient, secure, and flexible enough for a range of possible applications is not easy. Among such schemes, CL signature, proposed by Camenisch and Lysyanskaya [12], is one of the pairing-based signature schemes [8,10,12,24] that satisfies these conditions. It was widely used as a basic component in various cryptosystems such as anonymous credential systems, group signature, RFID encryption, batch verification signature, ring signature [2,3,5,11,12], as well as in aggregate signature [22].

Public-key aggregate signature (PKAS), introduced by Boneh, Gentry, Lynn, and Shacham [9], is a special type of PKS that enables anyone to aggregate different signatures generated by different signers on different messages into a short aggregate signature. Boneh et al. proposed the first full aggregate signature scheme in bilinear groups and proved its security in the random oracle model under the CDH assumption. After the introduction of aggregate signatures, various types of aggregate signatures such as sequential aggregate signatures [6,15,17–19] and synchronized aggregate signatures [1,14] were proposed. PKAS has numerous applications. In network and infrastructure: secure routing protocols, public-key infrastructure systems (signing certificate chains), sensor network systems, proxy signatures, as well as in applications: dynamically changing document composition (in particular, secure updating of software modules), secure transaction signing, secure work flow, and secure logs and repositories [1,6,7,9]. In all these applications, compressing the space consumed by signatures is the major advantage. Note that in the area of financial transactions, in particular, logs and repositories are very large due to regulatory requirements to hold records for long time periods. The effect of compressing signatures by aggregation increases with the number of data items; thus it is quite plausible that the financial sector may find variations of aggregate signature most useful.

Though PKAS can reduce the size of signers’ signatures by using the aggregation technique, it cannot reduce the size of signers’ public keys since the public keys are not aggregated. Thus, the total information the verifier needs to access is still proportional to the number of signers in the aggregate signature, since the verifier should retrieve all public keys of signers from a certificate storage. Therefore, it is very important to reduce the size of public keys. An ideal solution for this problem is to use identity-based aggregate signature (IBAS) that represents the public key of a signer as an identity string. However, IBAS requires a trust structure different from public key infrastructure, namely, the existence of an additional trusted authority, (the current IBAS schemes are in [6,14,15] and are all secure in the random oracle model.) To construct a PKAS scheme with short public keys, Schröder proposed a sequential aggregate signature scheme with short public keys based on the CL signature scheme [22]. In the scheme of Schröder, the public key consists of two group elements and the aggregate signature consists of four group elements, but the aggregate verification algorithm requires \( l \) pairing operations and \( l \) exponentiations where \( l \) is the number of signers in the aggregate signature. Therefore, this work, while nicely pointing at the