Abstract. Compact e-cash schemes allow a user to withdraw a wallet containing \( k \) coins in a single operation, each of which the user can spend unlikably. One big open problem for compact e-cash is to allow multiple denominations of coins to be spent efficiently without executing the spend protocol a number of times. In this paper, we give a \((\text{partial})\) solution to this open problem by introducing two additional protocols, namely, compact spending and batch spending. Compact spending allows spending all the \( k \) coins in one operation while batch spending allows spending any number of coins in the wallet in a single execution. We modify the security model of compact e-cash to accommodate these added protocols and present a generic construction. While the spending and compact spending protocol are of constant time and space complexities, complexities of batch spending is linear in the number of coins to be spent together. Thus, we regard our solution to the open problem as \( \text{partial} \). We provide two instantiations under the \( q\)-SDH assumption and the LRSW assumption respectively and present security arguments for both instantiations in the random oracle model.

Keywords: E-Cash, constant-size, compact, bilinear pairings.

1 Introduction

Electronic cash (e-cash) was invented by Chaum\[12\] in 1982. In its simplest form, an \( \text{e-cash} \) system consists of three parties (the bank \( \mathcal{B} \), the user \( \mathcal{U} \) and the shop \( \mathcal{S} \)) and four main procedures (account establishment, withdrawal, payment and deposit). The user \( \mathcal{U} \) first performs an account establishment protocol with the bank \( \mathcal{B} \). The currency circulating around is quantized as coins. \( \mathcal{U} \) obtains a coin by performing a withdrawal protocol with \( \mathcal{B} \) and spends the coin by participating in a spend protocol with \( \mathcal{S} \). To deposit a coin, \( \mathcal{S} \) performs a deposit protocol with \( \mathcal{B} \).

Security of e-cash refers to the fact that only the bank \( \mathcal{B} \) can produce a coin and for offline schemes, users who double-spent should be identified. The problem of double-spending only occurs in the electronic world due to easy duplication of digital coins. On the other hand, honest spenders cannot be slandered to have double spent \( \text{(exculpability)} \), and when the shops deposit the money from the payee, the bank should not be able to trace who the actual spender
is (anonymity). Many e-cash systems that provide the function of identifying double-spenders have been proposed, but most of them rely on a trusted third party (TTP) to revoke the anonymity so as to identify the double-spenders [7,17,11]. While the TTP cannot slander an honest user, its existence in fact implies that even honest users are not anonymous.

High efficiency is also of key importance for practical e-cash systems. For efficiency, we look at: (1) the time and bandwidth needed for the withdrawal, payment and deposit protocols; (2) the size of an electronic coin; and (3) the size of the bank’s database.

Camenisch, Hohenberger and Lysyanskaya [8] proposed a secure offline anonymous e-cash scheme (which we shall refer to as CHL scheme from now on) which is compact to address the efficiency issue. In their scheme, a wallet containing $k$ coins can be withdrawn and stored in complexity $O(\lambda + \log(k))$ for a security parameter $\lambda$, where each coin can be spent unlinkably with complexity $O(\lambda + \log(k))$ as well. Au et al. [3] construct compact e-cash from another approach by using a bounded accumulator. However, both schemes involve extensive use of proof-of-knowledge and the exact cost of each operation is somehow hard to quantify. One big open problem of compact e-cash, as stated in the CHL paper, is how to spend several coins in the wallet together efficiently.

**Related Results.** Compact e-cash scheme is closely related to $k$-TAA [20] and itself can be regarded as a multi-show credential system [13]. The main difference between a compact e-cash and a $k$-TAA is that in the former case, a token can only be used for a total of $k$ times while in the latter, a token can be shown for $k$-times to each application provider where $k$ is specified by each application provider independently. In some sense a $k$-TAA is more general. If the authentication of the $k$-TAA can be done non-interactively, that $k$-TAA scheme can be used as a compact e-cash system as follows. All shops play the role of a single application provider with $k$ being specified by the bank, while the bank plays the role of a GM. A user withdraws a coin by obtaining a credential from the bank and spend the coin by authenticating himself to the shop non-interactively. The shop deposits by submitting the authentication transcript back to the bank.

**Our Contributions.** Specifically, we make the following contributions

- We solve an open problem stated in the CHL paper by introducing the idea of compact spending and batch spending into compact e-cash systems.
- We present generic construction of compact e-cash system with these two added protocols and propose two instantiations
- We formalize a model to accommodate batch spending and compact spending protocols into compact e-cash schemes and present security arguments for our schemes.
- We outline how size of the wallet can be chosen arbitrarily by users while preserving user privacy during spending.