**INTRODUCTION**

**T**HE credit system is a platform that provides some form of credit evaluation for both individuals and non individual entities (e.g., organizations), which determines the “financial trustworthiness” of the individual and/or non individual entity [1], [2]. For instance, in the U.S., credit score is widely used in a broad range of applications, for example to determine whether an individual’s application for, say a credit card, home/automobile loan, etc., will be approved or an individual has to pay a higher insurance premium or higher interest rate (due to low credit score).

As shown in Fig. 1, a credit system generally comprises three types of participants (i.e., users, credit bureaus, and creditors). The user is a key part of the credit system, whose credit and loan activities (new account creation, account balance/credit card utilization, credit inquiries, and payment history) are reported to the credit bureaus. The latter is responsible for collecting, recording, and distributing relevant information (collectively referred to as “credit data”) about the user’s credit activities [3].

Such credit data are then requested by the creditors to compute the user’s credit score, which is then used to inform some decision-making. In other words, the credit score represent the credit value/risk/health of an individual or entity, which is a reference value for trust assessment [4]. The score is generated by analyzing the user’s credit data using an algorithm (i.e., risk model), a process known as credit score computation (CSC). Different models consider different factors and weights to compute the final credit score. For example, FICO scores are calculated based on the user’s payment history (35%), amounts owed (30%), length of credit history (15%), new credit (10%), and credit mix (10%).

1 There are different risk models in the literature, such as least squares support vector machines ensemble models for credit scoring [5], a measure of creditworthiness for sound financial decision-making [6], a partial credit model [7], and a fuzzy logistic regression based credit scoring model [8]. Using these models, creditors can take as input the credit data obtained from the bureaus and quickly obtain a credit score. However, these models do not consider the privacy protection of user credit data and their corresponding weights. That is, credit data in existing risk models are obtained directly by the creditors and the weightsin the model are all publicly known. This is a limiting factor in establishing a diversified and multilevel credit system.

User credit data can be mined for commercial/financial gains, for example by reselling such information to marketers and other business entities, or even cybercriminals. The data can then be mined to profile users (e.g., user online and shopping behaviors in order to provide targeted advertising), facilitate cybercriminal activities, such as identity theft, or identify individuals to facilitate I legal tracking and surveillance (e.g., by nation states) [9] [11]. Clearly, this is a topic of concern to most users. Also, depending on the application context, making weights used in the computation of the scores publicly know can also be abused by individuals or entities to game the system, for example to take a particular course of actions to enhance the credit score.

To protect the privacy of weights and credit data, the first thought is using a secure two-party computation [12], [13] It enables creditor and bureaus to evaluate a function (i.e., CSC formula) cooperatively wit out revealing to either party anything (e.g., weights and credit data) beyond the final credit score. For example, a simple function of computation based on weight computing is *f*(*k*1*, . . . , kt,m*1*, . . . , mt*) = Σ*t i*=1(*ki ·* *mi*), where *ki* and *mi* are weight and credit data, respectively. However, the existing two-party computation solutions, such as [14]–[16], generally do not consider the property of verification, meaning that a curious creditor (or bureaus) may provide fake weights (resp. credit data) to obtain each other’s privat information. Worse still, the creditor may lie about the final credit score to the bureaus or user. Thus, specific zero-knowledge proofs should be designed to further enhance the privacy of credit data and weights.

In this article, inspired by Goethals *et al.* [15], we also explore the potential of homomorphic encryption (HE) in the privacy-preserving design of CSC. Here, an HE scheme (e.g., [17], [18]) allows one to update the message in ciphertext without the need for decryption. That is, given encryption *E*(*k*1)*, . . . , E*(*kt*)*,E*(*m*1)*, . . . , E*(*mt*) of messages *k*1*, . . . , kt,m*1*, . . . , mt*, one can efficiently combine the ciphertext of *f*(*k*1*, . . , kt,m*1*, . . . , mt*), where *f*(*·*) is an efficiently computable function (mainly related to addition or multiplication operation in this article).

As mentioned earlier, a zero-knowledge proof tool [19]–[21] is required to achieve a privacy-enhancing CSC. Specifically, we need to prove three statements without revealing extra information, besides determining the validity of these statements. The first one is provided by creditor to state that ciphertexts are really corresponding to its weights and these weights are all in reasonable range, the second is for bureaus to prove that computed ciphertext is correctly embedded with suitable credit data, and the final is for creditor to prove that the obtained credit score is consistent with the ultimate ciphertext. Here, the reasonable range is to prevent creditor and bureaus from obtaining the other party’s private information (i.e., weights or credit data) via providing fake information.

In a typical real-world application, a noninteractive zeroknowledg (NIZK) proof (e.g., [22], [23]) tool is more practical, since we can avoid interactions and this reduces the communication cost. Thus, in this article, we first find an applicable HE scheme (i.e., Paillier encryption scheme [24], [25] with more efficient decryption algorithm) and then propose three NIZK schemes to support the design of a privacy-preserving CSC (PCSC) system.

In this article, we model a PCSC system designed to support the credit system. Unlike prior works that consider the design of risk models, we focus on its privacy protection instead Specifically, our main contributions are summarized as follows.

1) We propose the first formal description of a PCSC system alongside its security goals (i.e., weight confidentiality and cr dit confidentiality). This definition can be used in CSC, as well as other computations, such as digital asset settlement.

2) We introduce a concrete construction of PCSC, where the CSC is based on the weight. In the construction, we us Paillier encryption to hide the credit data and weights, and design three NIZK schemes to prove the validity of three statements mentioned earlier. Having introduced our key contributions above, we will now explain the layout of this article. Related preliminaries are presented in Section II. In Section III, we present the formal description of a PCSC system with its security goals, prior to presenting the concrete construction with security analysis in Section IV. In Section V, we evaluate the performance of our proposal.Finally, w conclude this article in Section VI.