Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 65 (2008) 57–74
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

Transforming semi-honest protocols to ensure accountability

Wei Jiang [a,*], Chris Clifton [a], Murat Kantarcıoğlu [b]

[a] Department of Computer Science, Purdue University, West Lafayette, IN 47907, United States
[b] Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75083, United States

Available online 26 July 2007

Abstract

The secure multi-party computation (SMC) model provides means for balancing the use and confidentiality of distributed data. This is especially important in the field of privacy-preserving data mining (PPDM). Increasing security concerns have led to a surge in work on practical secure multi-party computation protocols. However, most are only proven secure under the semi-honest model, and security under this adversary model is insufficient for many PPDM applications. SMC protocols under the malicious adversary model generally have impractically high complexities for PPDM. We propose an accountable computing (AC) framework that enables liability for privacy compromise to be assigned to the responsible party without the complexity and cost of an SMC-protocol under the malicious model. We show how to transform a circuit-based semi-honest two-party protocol into a protocol satisfying the AC-framework. The transformations are simple and efficient. At the same time, the verification phase of the transformed protocol is capable of detecting any malicious behaviors that can be prevented under the malicious model.

Keywords: Secure multiparty computation; Privacy-preserving

1. Introduction

Privacy and data utility are often perceived to be at odds. An omniscient data source would have many benefits, particularly in support of data mining. On the other hand, an omniscient data source eases misuse, such as the growing problem of identity theft. To prevent misuse of data, there has been a recent surge in laws mandating protection of confidential data, such as the European Community privacy standards [1], US healthcare laws [2], and California SB1386. However, this protection comes with a real cost through both added security expenditure and penalties and costs associated with disclosure. For example, CardSystems was terminated by Visa and American Express after having credit card information stolen [3]. ChoicePoint stock lost 20% of its value in the month following their disclosure of information theft. Such public relations costs can be enormous and could potentially kill a company. From lessons learned in practice, what we need is the ability to compute the desired "beneficial outcome" of sharing data for mining without having to actually

* Corresponding author.
E-mail addresses: wjiang@cs.purdue.edu (W. Jiang), clifton@cs.purdue.edu (C. Clifton), muratk@utdallas.edu (M. Kantarcıoğlu).

0169-023X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2007.06.014

share or disclose data. We can maintain the security provided by separation of control while still obtaining the benefits of a global data source.

Secure multi-party computation (SMC) [4–6] has recently emerged as an answer to this problem. Informally, if a protocol meets the SMC definitions, the participating parties learn only the final result and whatever can be inferred from the final result and their own inputs. A simple example is Yao's millionaire problem [5]: two millionaires want to learn who is richer without disclosing their actual wealth to each other. Recognizing this, the research community has developed many SMC protocols, for applications as diverse as forecasting [7], data analysis [8] and auctions [9].[1] With such a protocol, liability for disclosure of private information falls squarely on the original custodian of that information, as the data is not disclosed during the protocol and thus could not have been disclosed by other parties.

Formal definitions of SMC exist for two adversary models: semi-honest and malicious. In the semi-honest model, it is assumed that each party follows the protocol. However, after the protocol is complete, the adversary may attempt to compute additional information from the messages received during execution. In the malicious model, a party can diverge arbitrarily from normal execution of the protocol. It has been proven that for any polynomial-time algorithm, there exists a polynomial-time secure protocol that achieves the same functionality under either the semi-honest or the malicious model [4]. Nevertheless, most practical algorithms developed have only been proven secure under the semi-honest model. While not a proof, this certainly gives evidence that achieving security against a malicious adversary adds significant complexity and expense.

An SMC-protocol secure under the semi-honest model (or an SSMC-protocol) rarely provides sufficient security for practical applications. A dishonest party could learn private information by not following the protocol, then disclose that information, with blame falling on the innocent original data custodian. (Alternatively, the original data custodian could disclose the private data, then *claim* the other party was dishonest, learned and disclosed the data, and should share liability.) For example, two competing transportation companies want to mine useful patterns among their customers to decide if they can collaborate. Assume there exists an SSMC-protocol that searches for possible overlapping patterns. It is difficult to convince the companies of the need for the protocol if they trust each other; without trust (to follow the protocol correctly) a semi-honest protocol provides no guarantees. However, if cheating can be prevented or caught, contractual penalties can be used to overcome trust issues and enable collaboration. An SMC-protocol secure under the malicious model (or an MSMC-protocol) generally provides such a guarantee, but the complexity of an MSMC-protocol commonly prevents it from being adopted in practice.

Fortunately, our proposed AC-framework can be utilized to design more practical and efficient protocols. The idea behind the AC-framework is that a party who correctly followed the protocol can be proven to have done so and consequently prove that it did not know (and thus could not have disclosed) private data. This provides substantial practical utilities over a semi-honest protocol. In addition, although a malicious adversary participating in an AC-protocol may learn things that they should not know and damage the result, such a behavior could be detected under the AC-framework. Furthermore, since the AC-framework does not need to prevent disclosure to a malicious adversary, protocols can be less complex. In particular, much of the cost can be pushed to a verification phase which needs only be run to expose the culprit when disclosure is detected or auditing is performed to verify honest behaviors among collaborating parties. This enables protocols that approach the efficiency of semi-honest protocols and leads to many practical applications for which the semi-honest protocols are insufficient.

The goal of this paper is to show that without sacrificing its utility and efficiency, functionality computable under a two-party SSMC-protocol can be computed under the AC-framework. Although an MSMC-protocol directly prevents malicious behaviors, the verification phase of the AC-transformed protocol is at least able to detect any malicious behaviors that can be prevented under the malicious adversary model. The paper is organized as follows: Section 2 presents current state of the art from the literature of SMC. Section 3 introduces a simplified version of the AC-framework. Section 4 shows how to transform any SSMC-protocol to satisfy the simplified AC-framework based on certain techniques adopted in Pinkas' compiler. Section 5 provides an alternative transformation utilizing threshold homomorphic encryption. To demonstrate additional utilities

[1] We have only cited one early example of each.