SMC-MuSe: A Framework for Secure Multi-Party Computation on MultiSets

Georg Neugebauer[1], Ulrike Meyer[1], and Susanne Wetzel[2]
[1] Department of Computer Science, RWTH Aachen University
{neugebauer, meyer}@umic.rwth-aachen.de
[2] Department of Computer Science, Stevens Institute of Technology
swetzel@stevens.edu

Abstract: Secure Multi-Party Computation (SMC) offers a theoretically well-founded means to allow applications that preserve their users' privacy. We introduce SMC-MuSe, a framework for **S**ecure **Mu**lti-Party **C**omputation on **Mu**lti**Se**ts, which enables the privacy-preserving computation of set operations on multisets. SMC-MuSe is targeted to provide for the efficient implementation of specific interesting functions rather than on computing arbitrary ones. It is generic in the sense that it allows to compute any composition of privacy-preserving set intersections, unions, and reductions on multisets. The system model used in SMC-MuSe is kept close to the one assumed in theory and supports asynchronous communications, resilient SMC computations, and fully-automated key management.

1 Introduction

Today's Internet is full of applications through which users share private information with their friends and business partners (e.g., Doodle, Google Calendar, or Flickr). As a side effect, this information is shared with a trusted service provider. The majority of users may still be willing to trade the use of the free service for making their information available to the server. However, privacy concerns of users are rising and users become more suspicious with respect to the use of their (personal) information by service providers.

SMC offers a theoretically well-founded approach to resolve this issue. In general, it allows multiple parties to compute a function on their individual private inputs in a distributed fashion without revealing anything but the output of the function to each other or any server. In theory, SMC typically assumes three properties: (1) all participating parties can directly communicate with each other, (2) secure channels exist between each pair of parties, and (3) any other keying material required for the SMC protocol is pre-distributed.

Recently, several frameworks have been proposed that strive to bring SMC closer to practical applications (e. g., [DGKN09, BSMD10]). Most of these frameworks introduce trusted servers that carry out all SMC-related computations on encrypted inputs on behalf of the users. The problem of establishing secure channels (Property 2) and distributing additional keying material (Property 3) is deferred from the clients to the trusted servers. The computing servers need to be trusted by the users to correctly follow the protocol and do nothing but the required computations.

2 SMC-MuSe

SMC-MuSe is a carefully designed framework for secure multi-party computation on multisets in which all SMC-related computations are carried out on the clients themselves instead of on the server. In general, there are n parties $P_1, ..., P_n$, each holding a private input multiset $S_i\,(1 \leq i \leq n)$ chosen from a common domain D. The framework is targeted at the efficient computation of arbitrary compositions of intersections, unions, and element reductions of private multiset inputs [KS05] that can be expressed in the grammar

$$\Upsilon ::= S_i \mid Rd_t(\Upsilon) \mid \Upsilon \cap \Upsilon | S_i \cup \Upsilon | \Upsilon \cup S_i\,.$$

Applications range from e-Voting schemes and auctions to distributed network monitoring and scheduling applications [MNMW11]. SMC-MuSe offers a comprehensive support infrastructure to achieve *security, privacy, usability, platform independence, scalability, reliability, modularity, and simplicity*.

SMC-MuSe's system model addresses the three assumptions made in theoretical SMC. In particular, SMC-MuSe introduces two (non-colluding) server components: one that is solely responsible for relaying messages between clients and thus addresses Property 1 and one that automatically generates and distributes keying material to the involved clients (Property 3) and that additionally automates the setup of the secure channels (Property 2).

Table 1 compares SMC-MuSe with two existing SMC frameworks regarding prominent framework characteristics. All three frameworks are either based on *Shamir Secret Sharing* or *Additive Homomorphic Encryption*. VIFF's system model meets the properties in theoretical SMC. However, the setup, including the generation and distribution of keying material as well as the establishment of secure channels is cumbersome. SEPIA's system model introduces trusted servers which are responsible for all computations. The setup is still complex compared to SMC-MuSe. VIFF and SEPIA allow more generic computations but both frameworks permit less colluding attackers than SMC-MuSe. All three frameworks guarantee security against semi-honest adversaries.

We have demonstrated the potential of SMC-MuSe by implementing the Multi-Party Reconciliation on Ordered Sets (MPROS) protocols proposed in [NMW10, NBMW13]. These protocols allow multiple parties to find common inputs in their ordered input sets that maximize a common order on the intersection of their inputs. We have also built a privacy-preserving scheduling application for desktop and smartphone users on top of the MPROS protocols.

Characteristics	**VIFF**	**SEPIA**	**SMC-MuSe**
Cryptographic Primitive	SSS/AHE	SSS	AHE
Computable Functions	$Z_p, GF(2^8)$	Z_p	$Rd_t, \cap, \cup$
Setup phase	Manual	Semi-automatic	Automatic
Colluding attackers	$c < \frac{n}{2}$	$c < \frac{n}{2}$	$c \leq n-1$
Communication model	async.	sync.	async.
Security model	semi-honest	semi-honest	semi-honest

Table 1: Comparison of SMC-MuSe against two existing SMC frameworks

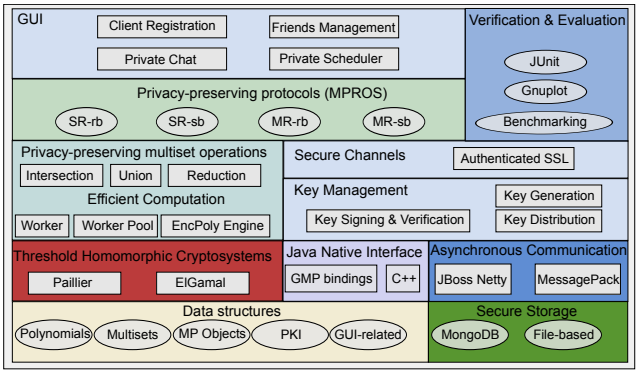

Figure 1: Overview of SMC-MuSe

SMC-MuSe is written in Java. The core library consists of two homomorphic cryptosystems, privacy-preserving multiset operations, multi-party reconciliation protocols, and different components for communication and computation. An overview of all implemented components is given in Figure 1. Our technical report [NM12] provides detailed information about secure multi-party computation on multisets, the framework design of SMC-MuSe, the system model, the design features, the MPROS protocols, the GUI components, the implementation and evaluation, and the comparison with other SMC frameworks.

Acknowledgments

This work has been supported by the DFG project ME 3704/1-1 and NSF Award CCF 1018616.

References

[BSMD10] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. SEPIA: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics. In *19th USENIX Security Symposium 2010*. USENIX, 2010.

[DGKN09] I. Damgard, M. Geisler, M. Kroigaard, and J. B. Nielsen. Asynchronous Multiparty Computation: Theory and Implementation. In *12th Public Key Cryptography 2009*. Springer Berlin Heidelberg, 2009.

[KS05] L. Kissner and D. X. Song. Privacy-Preserving Set Operations (Last mod. June 2006). In *25th Advances in Cryptology - CRYPTO 2005*. Springer Berlin Heidelberg, 2005.

[MNMW11] D. Mayer, G. Neugebauer, U. Meyer, and S. Wetzel. Enabling Fair and Privacy-Preserving Applications Using Reconciliation Protocols on Ordered Sets. In *34th IEEE Sarnoff Symposium*. IEEE, 2011.

[NBMW13] G. Neugebauer, L. Brutschy, U. Meyer, and S. Wetzel. Design and Implementation of Privacy-Preserving Reconciliation Protocols. In *6th Workshop on Privacy and Anonymity in the Information Society 2013*, ACM, 2013.

[NM12] G. Neugebauer and U. Meyer. SMC-MuSe: A Framework for Secure Multi-Party Computation on MultiSets. Technical report, Department of Computer Science, RWTH Aachen University, 2012.

[NMW10] G. Neugebauer, U. Meyer, and S. Wetzel. Fair and Privacy-Preserving Multi-Party Protocols for Reconciling Ordered Input Sets. In *13th Information Security Conference 2010*. Springer Berlin Heidelberg, 2010.