Secure Computation on Mobile Devices

Yan Huang
University of Virginia
yhuang@virginia.edu

Peter Chapman
University of Virginia
pchapman@cs.virginia.edu

David Evans
University of Virginia
evans@cs.virginia.edu

***Abstract*—Secure two-party computation has been an interesting theoretical concept for decades, but until recently has been considered too expensive for practical applications. With significant advances in the design and implementation of secure protocols, secure computation is becoming feasible in practice. However, with the notable exceptions of biometric identification and secure auctions, the research community has historically struggled to find realistic and useful applications for secure computation. In this work, we explore new opportunities implementing secure computing applications on rapidly evolving mobile computing platforms. Through a series of Android applications, we demonstrate the feasibility and power of secure computation protocols in this new domain.**

***Keywords*-secure computation; mobile devices; Yao's circuits.**

I. Introduction

Secure two-party computation enables two parties to evaluate a joint function without either revealing their inputs. In the 1980s, Yao developed an approach to compute any function securely through the use of garbled circuits [1]. The research community has historically viewed garbled circuits as too inefficient for practical applications, and thus turned to alternative approaches based on various application-specific properties of target functions for privacy and performance [2, 3]. Our recent [4] and ongoing work [5] has brought significant efficiency improvements in generating and evaluating garbled circuits, towards practical and useful privacy-preserving applications on mobile devices.

We choose to focus on mobile devices to demonstrate the feasibility of doing such computations on resource-limited devices, to show practical applications for such a technology, and to leverage the architecture of the mobile platform to address the often critiqued honest-but-curious threat model. Mobile applications offer a unique execution environment. Since these ubiquitous devices are highly portable, enforcing full physical control over them is much easier so people often trust them more than traditional computing devices such as desktops and laptops. Modern phones are personal, containing the most sensitive private information including phone calls, emails, contacts, and documents. Moreover, mobile devices are increasingly used in two-factor authentication schemes and even payment systems. We can leverage this to create privacy-preserving applications not feasible on traditional platforms. The challenge is that the processing power and memory available on mobile devices is still orders of magnitude below what is available on typical desktops, and the computation that can be done on battery-powered mobile devices is severely limited by the energy available.

New Trust Model. We will explore how operating on a mobile device frames the problem of secure computation. Implicit user-device and user-carrier trust relationships exist in this use-case. Trusting the device enables storing sensitive data while the carrier can often be regarded as a partially-trusted third party. The trust placed on carriers could be leveraged to address one notable weakness in the *semi-honest* adversary model. To that end, we will investigate how to use the carrier to ensure that all participating devices follow the protocol faithfully, but without leaking any information to the carrier.

Technical Challenges. For our implementation we have chosen the Android platform because of its open source license, healthy development community, widespread adoption, and support for Java development. Secure computation research experiments are traditionally conducted on desktop computers connected by a high-speed local area network. Early prototypes show that the limited processing power on mobile devices is a technical challenge that requires creative solutions through the optimization of algorithms and the construction of efficient garbled circuits. The availability of multi-core mobile devices will be useful with a parallelized approach which our current framework is designed to support. Furthermore, we hope to leverage the Renderscript API available in Android 3.0, which allows low-level, high performance execution on the device CPU or GPU to increase both concurrent operations and per instruction work [6]. Moreover, we are concerned about battery power, which is not an issue for desktops but often the most constrained resource for a mobile device.

II. Prior Work

We have developed a method to build privacy-preserving applications using garbled circuits that is much more efficient and scalable than previous implementations, most of which were built on Fairplay [7]. We have overcome the main impediment to practical applications of this technique in previous work, namely the huge amount of memory needed to store the entire circuit, by pipelining circuit generation and evaluation. We have also developed a set of

techniques for dramatically reducing the size of the garbled circuit needed to perform a computation, and demonstrated these techniques produce orders of magnitude performance improvements over the best previous techniques for several applications including privacy-preserving biometric identification [4, 5], genomic analysis based on edit distance and Smith-Waterman algorithms, private AES [5], and private set intersection [8].

These optimizations are accessible in a Java framework and library. Integration into existing systems is easy: a developer translates the secure components of the application into boolean circuits from which the library facilitates the secure computation.

III. Applications

Secure computation can enable a wide range of privacy-sensitive applications. Next, we describe interesting applications for mobile devices.

A. Set Intersection

We developed an efficient private set intersection protocol which enables a number of interesting application scenarios. For example, two people who meet at a conference could securely discover mutual contacts without revealing their address books, or two new acquaintances can find their common friends, hobbies, or places of interests without embarrassment as a result of privacy leaks.

Previous approaches to set intersection either use homomorphic encryption to evaluate a polynomial that encodes the set elements [9] or employ a secure encryption protocol. We developed a series of methods to compute set intersection more efficiently. For example, we developed a bit-vector-AND-based scheme, where each set element is mapped to a bit in the bit-vector, for computing the set intersection when the element space is relatively small (e.g. $\leq 2^{16}$). For larger element spaces (e.g. $\gg 2^{20}$), the AND-based scheme is impractical as the vector grows exponentially in the number of bits used to denote an element. Thus, we proposed a pair-wise comparisons based scheme, where element pairs are iteratively drawn from the two private input sets and fed into a secure equality test circuit. This scheme works well with sets of relatively small sizes (e.g. < 300), though the asymptotic cost is $O(n^2)$ where n is the size of the private sets. We also developed various more complex $O(n \log n)$ private set intersection schemes based on secure sorting networks which give superior performance when both the element space and set size are large. Depending on the parameters of a problem instance, we can choose the best solution from a rich spectrum of candidate schemes.

B. Genetic Analysis

In this scenario, we envision secure computing applications that allow friends or even strangers to search for kinship relationships (e.g., discovering you are likely to be third cousins) or estimate their off-springs' risks of hereditary diseases. Affordable services already exist to perform a computationally insecure genomic analysis [10, 11] by signing legal contracts. Many different ways exist to estimate the risks of hereditary diseases, varying in the level of complexity. In a simple model, genetic diseases are determined by recessive alleles, meaning that if both parents are carriers their children will suffer from that disease. This functionality can obviously be computed by AND-ing both parent's carrier bit. For more precise risk estimates, we are also developing an efficient circuit-based scheme that applies Mendel's law to compute phenotype probabilities from parent genotypes.

IV. Preliminary Results

We have developed prototypes to run private set intersection and genomic analysis on consumer Android devices. Preliminary results show that the mobile devices are roughly 1000 times slower than our desktop benchmarks. Fortunately, it still enables a good number of interesting secure mobile applications. We plan to improve the performance by customizing aspects of our implementation to better suit the mobile platform. We will demonstrate our applications using several different Android devices during the poster session.

V. Conclusion

Significant improvements in the efficiency of garbled circuit execution enable secure computation on mobile devices. Although performance hurdles remain, results from existing prototypes are promising, and the unique relationship between the device and the carrier may allow for new approaches to secure computation.

References

[1] A. Yao, "How to Generate and Exchange Secrets," in *Symposium on Foundations of Computer Science*, 1986.

[2] A. Jarrous and B. Pinkas, "Secure Hamming Distance Based Computation and Its Applications," in *Applied Cryptography and Network Security*, 2009.

[3] R. Gennaro, C. Hazay, and J. S. Sorensen, "Text Search Protocols with Simulation Based Security," in *Public Key Cryptography*, 2010.

[4] Y. Huang, L. Malka, D. Evans, and J. Katz, "Efficient Privacy-Preserving Biometric Identification," in *Network and Distributed System Security Symposium*, 2011.

[5] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster Secure Two-Party Computation Using Garbled Circuits," in *USENIX Security Symposium*, 2011.

[6] Android Developers, "Renderscript," http://developer.android.com/reference/android/renderscript/package-summary.html.

[7] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay—A Secure Two-party Computation System," in *USENIX Security Symposium*, 2004.

[8] Y. Huang, D. Evans, and J. Katz, "Private Set Intersection: Are Garbled Circuits Better than Custom Protocols?" Submitted to CCS 2011, prelimiary version available at http://www.cs.virginia.edu/yhuang/pubs/psi.pdf.

[9] M. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in *EUROCRYPT*, 2004.

[10] National Center for Biotechnology Information, "GeneTests," http://www.ncbi.nlm.nih.gov/sites/GeneTests/.

[11] KnowYourGenes.org, "Genetic Testing Information from the Genetic Disease Foundation," http://www.knowyourgenes.org/.