BOSCH

Invented for life

19th Crypto-Day

Corporate Research Security Group
Robert Bosch GmbH
November 14 and 15, 2013
Stuttgart, Germany

Day 1 – November 14, 2013

13:00 - 13:05 **Welcome**
13:05 - 13:50 **Keynote Talk: Security for Cyber-Physical Systems**
Harald Hönninger (VP Corporate Research, Robert Bosch GmbH)

Session 1

14:00 - 14:30 **Anonymes, verteiltes Microblogging für mobile Endgeräte**
Marius Senftleben (TU Darmstadt)
14:30 - 15:00 **Efficient Privacy Preserving Genome Processing Using ORAM Techniques**
Nikolaos P. Karvelas (TU Darmstadt)

15:00 - 15:30 Coffee Break

Session 2

15:30-16:00 **Cache Timing Attack on Scrypt**
Anne Barsuhn (Uni Weimar)
16:00-16:30 **Implementing Digital Certificates in Ada according to X.509**
Felix Trojan (Uni Weimar)

16:30-16:45 Coffee Break

Session 3

16:45 - 17:15 **The math behind algebraic geometry codes**
Moritz Scholz (Uni Gießen)
17:15 - 17:45 **Hardware-Assisted Ad-Hoc Secure Two-Party Computation on Smartphones**
Daniel Demmler (TU Darmstadt)

19:00 Dinner (Vapiano)

Day 2 – November 15, 2013

09:00-10:00 **Excursion to Bosch's Semiconductor Fab in Reutlingen**
The bus will start at 7:45 sharp at Feuerbach, gate 1.

11:30-12:15 **Keynote Talk: Product Security**
Jens Gramm (Bosch Centre of Competence Security, ETAS GmbH)

12:15-12:20 **Adjourn**

12:30-13:30 Common Lunch

Anonymes, verteiltes Microblogging für mobile Endgeräte

Marius Senftleben*, Mihai Bucicoiu*, Erik Tews*,
Frederik Armknecht†, Stefan Katzenbeisser* und Ahmad-Reza Sadeghi*

*CASED/TU Darmstadt †Universität Mannheim

Microblogging-Dienste wie Twitter sind vielgenutzt im Bereich Online Social Networking (OSN). Sie bieten den Nutzern die Möglichkeit Kurznachrichten von bis zu 140 Zeichen öffentlich zu versenden [KLP10] [SB13]. Eine zunehmende Verbreitung von mobilen Endgeräten wie Smartphones erlaubt den Nutzern auch unterwegs einen einfachen und benutzerfreundlichen Zugriff auf diese Dienste. Da die meisten Dienste auf zentralisierten Client-Server-Architekturen basieren und keine Verschlüsselung benutzen, sind sie im Hinblick auf Datenschutz verbesserungswürdig und die Zuverlässigkeit ist architekturbedingt angreifbar.

Wir präsentieren eine datenschutzfreundliche Microblogging-Architektur, die aus einem verteilten, dezentralisiertem Peer-to-Peer-Netzwerk von gleichberechtigten mobilen Anwendern besteht und zeigen zudem durch Simulationen die Praktikabilität der Architektur. Die Schutzziele der Architektur sind Sender-Anonymität, Privacy – worunter wir die Vertraulichkeit der Nachrichten und Gruppenmitgliedschaften verstehen – sowie Zensurresistenz. Dabei unterscheidet sich die Erreichung von Anonymität in dieser Architektur von anderen bekannten Ansätzen wie Crowds [RR98] und Mix-Netzwerken [Ch81]. In unserer Architektur versenden die mobile Nutzer mithilfe ihrer Smartphones verschlüsselte Gruppen-Nachrichten über Nahfunkschnittstellen und verbreiten dadurch die Nachrichten unter Gewährung der Vertraulichkeit. Die eingesetzte Verschlüsselung erlaubt es Chiffrate zu re-randomisieren, so dass diese unverkettbar werden. Dadurch sind ohne Kenntniss des Schlüssels keine Rückschlüsse auf die Gruppe, von der die Nachricht stammte, mehr möglich, und die ursprünglichen Sender der Nachrichten sind nicht mehr feststellbar.

Literatur

[KLP10] Kwak, H., Lee, C., Park, H., Moon, S. What is twitter, a social network or a news media? *Proceedings of the 19th international conference on World wide web*, pp. 591–600 (2010).

[SB13] Statistic Brain: Twitter statistics. `www.statisticbrain.com/twitter-statistics/` (Juli 2013).

[RR98] Reiter, M.K., Rubin, A.D.: Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.* 1(1), 66–92 (November 1998).

[Ch81] Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* 24(2) (February 1981).

Efficient Privacy Preserving Genome Processing Using ORAM Techniques

Nikolaos P. Karvelas*, Andreas Peter**, Stefan Katzenbeisser* and Sebastian Biedermann*

* TU Darmstadt & CASED ** University of Twente

Recent developments in genomics enable new business and research models having many individuals' genomes as an essential driving force. It is foreseeable that this genomic data will be outsourced to a central service provider (e.g. the cloud), thus allowing different applications, such as personalized medicine, large-scale genomic research, disease susceptibility etc.

We present a novel mechanism that allows a patient to securely outsource her genomic data to the cloud while at the same time to delegate to an investigator (e.g. a physician) the right to run certain algorithms (e.g. medical tests) on her data. The mechanism is privacy-preserving, meaning that the investigator only learns the result of his algorithm on the patient's genome, while the cloud learns nothing at all. Our protocol only requires the patient to be online for the delegation of rights and is thereafter completely non-interactive with the patient.

Despite the big size of genomic sequences, we achieve reasonable efficiency by dividing the patient's genome into small blocks (e.g. SNPs) on which we can run efficient secure multiparty protocols. Our main technical contribution is a new ORAM-like construction to hide the access patterns on these blocks. Our mechanism is based on Williams et al.'s ORAM [WST12], adjusted in a way that allows the outsourcing of its most expensive operation (the "reshuffle") to an independent proxy without loss of privacy.

The delegation of rights to the investigator presents itself as a major challenge since he is not allowed to decrypt retrieved blocks. Such a setting has not been considered in previous ORAM constructions. In our work, we solve this issue by utilizing the independent proxy and a combination of homomorphically encrypted Bloom filters and secure multiparty computation techniques.

Processing the encrypted data is done once it has been retrieved from the cloud. The investigator transforms the function (this can be *any* function, e.g., some medical test) which he wants to evaluate the data on into a circuit, using the compiler presented in [HFKV12]. The output circuit he then passes to the proxy server, who runs a secure two-party computation protocol with the cloud and sends him the result back. Employing the compiler of [HFKV12] has the advantage that the investigator does not need any particular cryptographic expertise to transform his function into a secure protocol. This facilitates the use of medical tests that are to be developed in the future.

References

[HFKV12] Andreas Holzer, Martin Franz, Stefan Katzenbeisser, and Helmut Veith. Secure two-party computations in ansi c. In CCS'12. ACM, 2012.

[WST12] Peter Williams, Radu Sion, and Alin Tomescu. Privatefs: a parallel oblivious file system. In CCS'12. ACM, 2012.

Cache Timing Attack on Scrypt

Anne Barsuhn, Stefan Lucks, Christian Forler

Bauhaus-Universität Weimar,
Germany
anne.barsuhn@uni-weimar.de

Brute-force attacks have become very effective nowadays. It is not enough anymore to make a cryptographic key derivation function as slow as possible without compromising its usability. Since the introduction of GPU clusters parallelised brute-force attacks are not affected by this anymore. Therefore, Colin Percival developed a sequential memory-hard key derivation algorithm called `scrypt` [1]. `scrypt` uses the password to build a large array full of hashes which takes up about 1 Mb of memory or more. Those hash entries are used to compute the new key. With an index derived from the original key `scrypt` accesses the hash array in a pseudo-random pattern. Thus making the memory-access pattern of `scrypt` password dependent which results in a vulnerability against cache-timing attacks. Although the idea of memory-hard key derivation functions was progressive this seemed to be the weak point of `scrypt`.
A cache-timing attack on `scrypt` was introduced in [2]. This work is concerned with implementing this cache-timing attack based on known AES cache attacks. Different methods were analysed with respect to their effectiveness on modern hardware.
Since new processors are implemented with a lot of optimization techniques it was crucial to find a way to bypass optimization and deduce information from time leaks. This was accomplished by using the algorithm from [3] and modifying it to fit the attack from [2].

References

[1] Colin Percival. Stronger Key Derivation via Sequential Memory-Hard Functions. presented at BSDCan'09, May 2009, 2009.

[2] Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In *FSE*, pages 196–215, 2012.

[3] Eran Tromer, Dag Arne Osvik, and Adi Shamir. Efficient Cache Attacks on AES, and Countermeasures. *J. Cryptology*, 23(1):37–71, 2010.

Implementing Digital Certificates in Ada according to X.509

Felix Trojan, Martin Trenkmann, Christian Forler

Bauhaus-Universität Weimar,
Germany
{Felix.Trojan, Martin.Trenkmann, Christian.Forler}@uni-weimar.de

Almost all web-based internet services use *Secure Sockets Layer (SSL)* to secure their connections. SSL again depends on a *Public Key Infrastructure (PKI)* defined by the X.509 standard [3]. Most Web servers and browsers are expected to support X.509, because most of the web depends on it. But only a small variety of tools are available, when it comes to using and managing X.509 certificates. Most of of them are proprietary, which is often not desired by modern security standpoints. The only open source tool that is commonly used is the OpenSSL Project. Besides that, OpenSSL has a long history of security vulnerabilities[1], with a lot of them related to encoding issues. Being written in C, OpenSSL is very prone to programming errors. Therefore a more strict and safety-critical language, like Ada, would be more suited. The goal of this master thesis is the implementation of a X.509 library written in Ada, to provide an alternative to OpenSSL and to furthermore give Ada developers a corner stone to develop their own applications using X.509 certificates.
The data structures for X.509 are defined in the Abstract Syntax Notation One (ASN.1) defined by X.680-X.699. But it is common practice to use the ASN.1 specifications for X.509 that are defined by *RFC5280* [1]. Furthermore rules for encoding and decoding are provided by the ASN.1 Encoding Rules which are defined in X.690 [2]. Usually an ASN.1 toolkit is used to translate the ASN.1 data structures into data structures of the target language. This toolkit also provides a codec that implements the ASN.1 Encoding Rules so that the data can be transmitted in a common binary representation. For the most common languages like C, C++ and Java such toolkits exist, but for the majority of other programming languages they do not. Fortunately a toolkit called `Ada ASN.1 Toolkit (AATK)` was developed by Martin Trenkmann at our university. It provides an ASN.1 to Ada compiler. The extension of `AATK` with a codec that follows the ASN.1 Encoding Rules was also part of this work. For cryptographic algorithms used with X.509 we also use a library developed at the Bauhaus-Universität: the *Ada Crypto Library (ACL) by Christian Forler*. In addition, a variety of tests ensure the correctness of the encoding and the X.509 implementation.

References

[1] Cooper, D. and Santesson, S. and Farrell, S. and Boeyen, S. and Housley, R. and Polk, W. RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.

[2] ITU-T: Recommendation X.690, Information technology - ASN.1 encoding rules Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), July 2002.

[3] ITU-T: Recommendation X.509, Information technology - Open systems interconnection - The Directory: Public-key and attribute certificate frameworks, November 2008.

[1] `http://www.openssl.org/news/vulnerabilities.html`

The math behind algebraic geometry codes

Moritz Scholz
Mathematisches Institut
Justus-Liebig-Universität Gießen
Germany

In the late 1970s, V.D. Goppa discovered in [Gop81] that divisors of algebraic function fields can be used to construct a large class of codes which nowadays are known as algebraic geometry codes (AG codes). Such well-known codes as BCH, Reed-Solomon and Goppa codes can be defined using AG codes. In order to define AG codes in general and prove their very good properties, one needs to dive into the theory of algebraic function fields and places (or rational points on algebraic curves, if a geometric viewpoint is desired; see [Pre03]). The main result which is needed to study AG codes is the famous Riemann-Roch Theorem which can be applied to the theory of function fields. The talk provides an insight into the rich theory of algebraic function fields up to and including the Riemann-Roch Theorem and demonstrates how to apply these results to prove some basic properties of AG codes.

References

[Gop81] V.D. Goppa. Codes on algebraic curves. *Soviet Math. Dokl.*, 24 No. 1:170 – 172, 1981.

[Pre96] Oliver R. Pretzel. *Error correcting codes and finite fields.* Oxford applied mathematics and computing science series. Clarendon Press, Oxford, student edition, 1996.

[Pre03] Oliver R. Pretzel. *Codes and Algebraic Curves.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2003.

[Roc65] G. Roch. Ueber die Anzahl der willkürlichen Constanten in algebraischen Functionen. *Journal für die reine und angewandte Mathematik*, 64:372ff, 1865.

[Sti09] Henning Stichtenoth. *Algebraic Function Fields and Codes.* Springer Graduate Texts in Mathematics, second edition, 2009.

Hardware-Assisted Ad-Hoc Secure Two-Party Computation on Smartphones (Extended Abstract)

Daniel Demmler and Thomas Schneider and Michael Zohner
Engineering Cryptographic Protocols Group,
EC SPRIDE, Technische Universität Darmstadt, Germany

Secure computation allows multiple mutually distrusting parties to jointly compute a function on their private inputs without revealing anything but the function output. This technique is particularly interesting in the context of mobile devices, such as smartphones, where typically highly sensitive user data is stored and processed. The protection of this data is desirable but challenging, due to the computational complexity of secure computation protocols and the limited processing power, memory and battery-life of mobile hardware.

In this work we focus on the practical realization of secure two-party computation in a mobile environment and the possibility of enhancing it by using a trusted hardware token.

Previous work in the field, e.g. [HCE11], is mainly based on Yao's garbled circuit protocol [Yao86]. Further protocols employ homomorphic encryption, but are usually limited to a specific use case. We follow a different and more efficient approach by implementing the protocol by Goldreich-Micali-Wigderson [GMW87] and allowing generic functions to be evaluated. This is done on off-the-shelf Android smartphones using a general purpose microSD smartcard.

To address the aforementioned performance issues, we shift the most expensive cryptographic operations into an initialization phase on the trusted hardware token. This phase is independent of the function to be evaluated and can be executed locally when the phone is idle, e.g., charging overnight. The pre-generated trusted data enables us to efficiently mask sensitive user data, thus minimizing the ad-hoc computation time. For the purpose of securely distributing this data from the trusted token to the user, we implemented two secure channel protocols on the smartcard.

Our proof-of-concept implementation allows for managing the hardware token and running ad-hoc secure two-party computation with several use cases: private set intersection in order to find common contacts or securely scheduling a meeting, optionally with location preferences. However, our implementation is generic, i.e., new functionalities can easily be introduced by providing the Boolean circuit to be evaluated securely. The performance of our mobile implementation is extensively evaluated and found to be able to compete with state-of-the-art protocols implemented on desktop PCs [HEK12].

References

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, USA, 1987. ACM.

[HCE11] Yan Huang, Peter Chapman, and David Evans. Privacy-preserving applications on smartphones. In *Proceedings of the 6th USENIX conference on Hot topics in security*, HotSec'11, USA, 2011. USENIX Association.

[HEK12] Y. Huang, D. Evans, and J. Katz. Private set intersection: Are garbled circuits better than custom protocols? In *Network and Distributed Security Symposium (NDSS'12)*. The Internet Society, 2012.

[Yao86] A. C. Yao. How to generate and exchange secrets. In *Foundations of Computer Science (FOCS'86)*, pages 162–167. IEEE, 1986.