Timing attacks in security protocols: symbolic framework and proof techniques*

Vincent Cheval[1] and Véronique Cortier[2]

[1] School of Computer Science, University of Birmingham, UK
[2] LORIA, CNRS, France

Abstract. We propose a framework for timing attacks, based on (a variant of) the applied-pi calculus. Since many privacy properties, as well as strong secrecy and game-based security properties, are stated as process equivalences, we focus on (time) trace equivalence. We show that actually, considering timing attacks does not add any complexity: time trace equivalence can be reduced to length trace equivalence, where the attacker no longer has access to execution times but can still compare the length of messages. We therefore deduce from a previous decidability result for length equivalence that time trace equivalence is decidable for bounded processes and the standard cryptographic primitives.
As an application, we study several protocols that aim for privacy. In particular, we (automatically) detect an existing timing attack against the biometric passport and new timing attacks against the Private Authentication protocol.

1 Introduction

Symbolic models as well as cryptographic models aim at providing high and strong guarantees when designing security protocols. However, it is well known that these models do not capture all types of attacks. In particular, most of them do not detect *side-channel* attacks, which are attacks based on a fine analysis of *e.g.*, time latencies, power consumption, or even noise [26,27,9]. The issue of side-channel attacks is well-known in cryptography. One of prominent issues is the fact that many cryptosystems such as RSA or Difie-Hellman involve modular exponentiation. To remain efficient, modular exponentiation is implemented using (variants of) the fast exponentiation algorithm. Basically, this algorithm requires one squaring for each bit equals to 0 in the key, and one squaring and one multiplication for each bit equals to 1 in the key. A fine observation of the computation time or the power consumption leaks information on the secret key. Of course, counter-measures have been proposed but many variations of the attack remain.

The same kind of issues occur at the protocol level as well. For example, the biometric passport contains an RFID chip that stores sensitive information

* The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement number 258865 (ERC ProSecure project).

such as the name, nationality, date of birth, etc. To protect users' privacy, data are never sent in the clear. Instead, dedicated protocols ensure that confidential data are sent encrypted between the passport and the reader. However, a minor variation in the implementation of the protocol in the French passport has led to a privacy flaw [7]. Indeed, by observing the error message when replaying some old message, an attacker could learn whether a given passport belongs to Alice or not. The attack has been fixed by unifying the error messages produced by the passports. However, it has been discovered [18] that *all* biometric passports (from all countries) actually suffer from exactly the same attack as soon as the attacker measures the computation time of the passport instead of simply looking at the error messages.

The goal of the paper is to provide a symbolic framework and proof techniques for the detection of timing attacks on security protocols. Symbolic models for security protocols typically assume "the perfect encryption hypothesis", abstracting away the implementation of the primitives. We proceed similarly in our approach, assuming a perfect implementation of the primitives w.r.t. timing. It is well known that implementation robust against side-channel attacks should, at the very least, be "in constant time", that is, the execution time should only depend on the number of blocks that need to be processed. "Constant time" is not sufficient to guarantee against timing attacks but is considered to be a minimal requirement and there is an abundant literature on how to design such implementations (see for example the NaCl library [1] and some related publications [25,12]). One could think that side-channel attacks are only due to a non robust implementation of the primitives and that it is therefore enough to analyze in isolation each of the cryptographic operations. However, in the same way that it is well known that the perfect encryption assumption does not prevent flaws in protocols, a perfect implementation of the primitives does not prevent side-channel attacks. This is exemplified by the timing attack found against the biometric passport [18] and the timing attacks we discovered against the Private Authentication protocol [5] and several of its variants. These attacks require both an interaction with the protocol and a dedicated time analysis. Robust primitives would not prevent these attacks.

Our first contribution is to propose a symbolic framework that models timing attacks at the protocol level. More precisely, our model is based on the applied-pi calculus [3]. We equip each function symbol with an associated time function as well as a length function. Indeed, assuming a perfect implementation of the primitives, the computation time of a function typically only depends on the size of its arguments. Each time a process (typically a machine) performs an observable action (*e.g.*, it sends out a message), the attacker may observe the elapsed time. Our model is rather general since it inherits the generality of the applied-pi calculus with *e.g.*, arbitrary cryptographic primitives (that can be modeled through rewrite systems), possibly arbitrarily replicated processes, etc. Our time and length functions are also arbitrary functions that may depend on the machine on which they are run. Indeed, a biometric passport is typically much slower than a server. Moreover, a server usually handles thousands of requests at the same time, which prevents from a fine observation of its computation time. Our model is flexible enough to cover all these scenarios. Finally, our model covers more than just timing attacks. Indeed, our time functions not only model execution times but also any kind of information that can be leaked by the execution, such as power consumption or other "side-channel" measurements.

Our second main contribution is to provide techniques to decide (time) process equivalence in our framework. Equivalence-based properties are at the heart of many security properties such as privacy properties [21,7] (*e.g.*, anonymity, unlinkability, or ballot privacy), strong secrecy [15] (*i.e.* indistinguishability from random), or game-based security definitions [4,20] (*e.g.*, indistinguishability from an ideal protocol). Side channel attacks are particularly relevant in this context where the attacker typically tries to distinguish between two scenarios since any kind of information could help to make a distinction. Several definitions of equivalence have been proposed such as trace equivalence [3], observational equivalence [3], or diff-equivalence [14]. In this paper, we focus on trace equivalence. In an earlier work [17], we introduced length (trace) equivalence. It reflects the ability for an attacker to measure the length of a message but it does not let him access to any information on the internal computations of the processes.
Our key result is a generic and simple simplification result: time equivalence can be reduced to length equivalence. More precisely, we provide a general transformation such that two processes P and Q are in time equivalence if and only if their transformation $\tilde{P}$ and $\tilde{Q}$ are in length equivalence.

$$P \approx_{\mathsf{ti}} Q \Leftrightarrow \tilde{P} \approx_{\ell} \tilde{Q}$$

This result holds for an arbitrary signature and rewriting system, for arbitrary processes - including replicated processes, and for arbitrary length and time functions. The first intuitive idea of the reduction is simple: we add to each output a term whose length encodes the time needed for the intermediate computations. The time elapsed between two outputs of the same process however does not only depend on the time needed to compute the sent term and the corresponding intermediate checks. Indeed, other processes may run in parallel on the same machine (in particular other ongoing sessions). Moreover, the evaluation of a term may fail (for example if a decryption is attempted with a wrong key). Since we consider else branches, this means that an else branch may be chosen after a failed evaluation of a term, which execution time has to be measured precisely. The proof of our result therefore involves a precise encoding of these behaviors.

A direct consequence of our result is that we can inherit existing decidability results for length equivalence. In particular, we deduce from [17] that time equivalence is decidable for bounded processes and a fixed signature that captures all standard cryptographic primitives. We also slightly extend the result of [17] to cope with polynomial length functions instead of linear functions.

As an application, we study two protocols that aim for privacy: the private authentication protocol (PA) [5] and the Basic Authentication Protocol (BAC) of the biometric passport [2]. Using the APTE tool [16] dedicated to (length) trace equivalence, we retrieve the flaw of the biometric passport mentioned earlier. We demonstrate that the PA protocol is actually not private if the attacker can measure execution times. Interestingly, several natural fixes still do not ensure privacy. Finally, we provide a fix for this protocol and (automatically) prove privacy.

Related work. Several symbolic frameworks already include a notion of time [11,22,19,23,24]. The goal of these frameworks is to model timestamps. The system is given a global clock, actions take some number of "ticks", and participants may compare time values. Depending on the approach, some frameworks (e.g. [11,22]) are analysed using interactive theorem provers, while some others (e.g. [19,24]) can be analysed automatically using for example time automata techniques [24]. Compared to our approach, the representation of time is coarser: each action takes a fixed time which does not depend on the received data while the attack on e.g. the biometric passport precisely requires to measure (and compare) the time of a given action. Moreover, these frameworks consider trace properties only and do not apply to equivalence properties. They can therefore not be applied to side-channel analysis.

On the other hand, the detection or even the quantification of information possibly leaked by side-channels is a subject thoroughly studied in the last years (see e.g. [28,10,30,13,8]). The models for quantifying information leakage are typically closer to the implementation level, with a precise description of the control flow of the program. They often provide techniques to *measure* the amount of information that is leaked. However, most of these frameworks typically do not model the cryptographic primitives that security protocols may employ. Messages are instead abstracted by atomic data. [28] does consider primitives abstracted by functions but the framework is dedicated to measure the information leakage of some functions and does not apply to the protocol level. This kind of approaches can therefore not be applied to protocols such as BAC or PA (or when they may apply, they would declare the flawed and fixed variants equally insecure).

Fewer papers do consider the detection of side-channel attacks for programs that include cryptography [29,6]. Compared to our approach, their model is closer to the implementation since it details the implementation of the cryptographic primitives. To do so, they over-approximate the ability of an attacker by letting him observe the control flow of the program, e.g. letting him observe whether a process is entering a then or an else branch. However privacy in many protocols (in particular for the BAC and PA) precisely relies on the inability for an attacker to detect whether a process is entering a then (meaning e.g. that the identity is valid) or an else branch (meaning e.g. that the identity is invalid). So the approach developed in [29,6] could not prove secure the fixed variants of BAC and PA. Their side-channel analysis is also not automated, due to the expressivity of their framework.

References

1. http://nacl.cr.yp.to/.

2. Machine readable travel document. Technical Report 9303, International Civil Aviation Organization, 2008.
3. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *28th ACM Symp. on Principles of Programming Languages (POPL'01)*, 2001.
4. M. Abadi and A. Gordon. A calculus for cryptographic protocols: The spi calculus. In *4th Conference on Computer and Communications Security (CCS'97)*, pages 36–47. ACM Press, 1997.
5. Martín Abadi and Cédric Fournet. Private authentication. *Theoretical Computer Science*, 322(3):427–476, 2004.
6. Jos Bacelar Almeida, Manuel Barbosa, Gilles Barthe, and François Dupressoir. Certified computer-aided cryptography: Efficient provably secure machine code from high-level implementations. In *21st ACM Conference on Computer and Communications Security (CCS'13)*, 2013.
7. M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *23rd IEEE Computer Security Foundations Symposium (CSF'10)*, 2010.
8. Michael Backes, Goran Doychev, and Boris Köpf. Preventing side-channel leaks in web traffic: A formal approach. In *Network and Distributed System Security Symposium (NDSS'13)*, 2013.
9. Michael Backes, Markus Duermuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. Acoustic emanations of printers. In *19th USENIX Security Symposium*, 2010.
10. Michael Backes, Boris Köpf, and Andrey Rybalchenko. Automatic discovery and quantification of information leaks. In *Symposium on Security and Privacy (S&P'09)*, 2009.
11. G. Bella and L. C. Paulson. Kerberos version IV: Inductive analysis of the secrecy goals. In *5th European Symposium on Research in Computer Security (Esorics'98)*, volume 1485 of *LNCS*. Springer, 1998.
12. Daniel J. Bernstein, Tung Chou, and Peter Schwabe. Mcbits: Fast constant-time code-based cryptography. In *Cryptographic Hardware and Embedded Systems (CHES 2013)*, volume 8086 of *Lecture Notes in Computer Science*, pages 250–272. Springer, 2013.
13. Fabrizio Biondi, Axel Legay, Pasquale Malacaria, , and Andrzej Wasowski. Quantifying information leakage of randomized protocols. In *14th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'13)*, 2013.
14. B. Blanchet, M. Abadi, and C. Fournet. Automated Verification of Selected Equivalences for Security Protocols. In *20th Symposium on Logic in Computer Science*, pages 331–340, 2005.
15. Bruno Blanchet. Automatic proof of strong secrecy for security protocols. In *Symposium on Security and Privacy (S&P'04)*, pages 86–100. IEEE Comp. Soc. Press, 2004.
16. Vincent Cheval. Apte: an algorithm for proving trace equivalence. In Erika Ábrahám and JKlaus Havelund, editors, *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14)*, Lecture Notes in Computer Science, Grenoble, France, April 2014. Springer. to appear.
17. Vincent Cheval, Véronique Cortier, and Antoine Plet. Lengths may break privacy – or how to check for equivalences with length. In *25th International Conference on Computer Aided Verification (CAV'13)*, volume 8043 of *Lecture Notes in Computer Science*, pages 708–723. Springer, 2013.

18. Tom Chothia and Vitaliy Smirnov. A traceability attack against e-passports. In *14th International Conference on Financial Cryptography and Data Security*, 2010.
19. E. Cohen. Taps: A first-order verifier for cryptographic protocols. In *13th IEEE Computer Security Foundations Workshop (CSFW00)*. IEEE Computer Society, 2000.
20. H. Comon-Lundh and V. Cortier. Computational soundness of observational equivalence. In *15th Conf. on Computer and Communications Security (CCS'08)*, 2008.
21. Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, (4):435–487, July 2008.
22. N. Evans and S. Schneider. Analysing time dependent security properties in csp using pvs. In *6th European Symposium on Re- search in Computer Security (Esorics'00)*, page 222237. Springer-Verlag, 2000.
23. R. Gorrieri, E. Locatelli, and F. Martinelli. A simple language for real-time cryptographic protocol analysis. In *12th European Symposium on Programming (ESOP'03)*, page 114128, 2003.
24. Gizela Jakubowska and Wojciech Penczek. Modelling and checking timed authentication of security protocols. *Fundamenta Informaticae*, pages 363–378, 2007.
25. Emilia Käsper and Peter Schwabe. Faster and timing-attack resistant aes-gcm. In *Cryptographic Hardware and Embedded Systems (CHES 2009)*, volume 5747 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2009.
26. Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '96)*, pages 104–113. Springer-Verlag, 1996.
27. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*, pages 388–397. Springer-Verlag, 1999.
28. Boris Köpf and David Basin. An information-theoretic model for adaptive side-channel attacks. In *14th ACM Conference on Computer and Communications Security (CCS'07)*, 2007.
29. David Molnar, Matt Piotrowski, David Schultz, and David Wagner. The program counter security model: Automatic detection and removal of control-flow side channel attacks. In *International Conference and Information Security and Cryptology (ICISC'05)*, pages 156–168, 2005.
30. Quoc-Sang Phan, Pasquale Malacaria, Oksana Tkachuk, and Corina S. Pasareanu. Symbolic quantitative information flow. In *ACM SIGSOFT Software Engineering Notes*, 2012.