

Final Report of Grant GR/S98139/01: Automated Analysis of Security Critical Systems

Graham Steel Alan Bundy Jacques Fleuriot

October 5, 2007

1 Background/Context

The aim of the project was to investigate the application of security protocol analysis techniques to security APIs. Security protocols are short programs that describe the secure exchange of information over an insecure network, using cryptography. Security APIs are the Application Program Interfaces of tamper-resistant hardware security modules, commonly used in security critical applications such as cash machines (ATMs). Security protocol analysis has been the subject of intensive research by the formal methods community over the last 20 years, but the use of these techniques for analysing APIs had not been attempted prior to our proposal. In the early 2000s, a number of powerful attacks on APIs currently in use in the banking network were discovered as the result of painstaking manual analysis by expert researchers. Hence the aim of this project was to develop techniques to automate the analysis of APIs, taking security protocol analysis as a starting point.

2 Key Advances and Supporting Methodology

Although there are many similarities between security APIs and security protocols, some important differences both in the nature of the commands used, and the attacks known to be of interest, meant that protocol analysis tools were not immediately applicable to the problem of API analysis. The main scientific advances of the project were in developing or adapting techniques suitable for API analysis. In this section, we first describe these advances. We then describe how these new techniques were used in a number of case studies. Finally, we compare these achievements to the original objectives of the project.

2.1 Scientific Advances

Key advances include the following:

1. **Algebraic properties of cryptofunctions.** In the classical Dolev-Yao (DY) model for protocol analysis, bitstrings are abstracted to logical terms, and cryptographic functions such as encryption and decryption are considered as functions on those terms. Only identical terms are considered equal in the model. In real systems, cryptographic functions usually have algebraic properties that result in a combinatorial explosion in the model. We have proposed novel ways of dealing with the combinatorial possibilities introduced by the bitwise XOR operation, [10], resulting in decidability results (in collaboration with Véronique Cortier, [5]), and a representation of terms which leads to an efficient decision procedure, [4].
2. **Covert channel attacks.** A significant family of API attacks uses informed online guessing to determine the value of a customer's PIN. This kind of attack is completely outside the scope of DY modelling; such an intruder model had never been formalised before. In perhaps the most novel work in the project, we proposed the use of a Markov decision process model to account for the intruder's choice of actions and the probabilistic branching based on unknown PIN

values. We showed how such models could be automatically generated from API specifications using constraint logic programming, and analysed to determine the most effective attack using a probabilistic model checker, [12].

3. **Key conjuring.** This is the process by which a would-be attacker obtains an encrypted key for a tamper-proof device by repeatedly trying random values. This attack technique has been shown to be effective against a number of APIs, but had not been formally treated before. In collaboration with Véronique Cortier and Stéphanie Delaune (INRIA Nancy) we defined a formal transformation which identifies computationally feasible key conjuring operations and allows them to be incorporated into a DY model, and proved further decidability results, [2].
4. **Parallel Key Search.** Together with Judicaël Courant (Verimag Grenoble), we have begun to develop a formal model for parallel key search, another non-DY attack method. Early work was presented at the FCC workshop this summer, [15], and follow-up work is ongoing, in collaboration with Mathieu Baudet (DCSSI).
5. **Modelling state in API analysis.** Some APIs (such as PKCS#11, see §2.2) have mutable global state (the attributes of the stored keys), which must be accounted for in order to obtain accurate models. We developed techniques for accounting for this in existing protocol analysis models, using constraints.

2.2 Case Studies

We applied the techniques we developed to the following APIs as case studies:

- (i) **IBM 4758 CCA API.** The IBM 4758 CCA is an important example of a security API, widely deployed in the ATM network. It uses IBM's patented method for key management, which makes extensive use of the bitwise XOR operation. The key management API was analysed using a method for modelling APIs as protocols, with the help of the CL-AtSe tool. A new vulnerability was discovered and reported to IBM, resulting in a warning in the subsequent release of the developer's manual. We proposed fixes, and using the decision procedure mentioned in advance 1 (above), we verified our fixes to be sound for an unbounded number of API command calls. A summary of the results appeared in [4], and a journal paper on the work is under review, [8]. The PIN processing commands of the API were analysed using AnaBlock, our tool that implements the developments in advance 2 (above). Results appeared in [12], including a previously unknown vulnerability in the IBM PIN verification function.
- (ii) **nCipher payShield** The PIN processing functions of the nCipher payShield were analysed using AnaBlock, and the results reported to nCipher in [9].
- (iii) **PKCS#11.** The RSA PKCS#11 API is widely used in a variety of different tamper-proof devices, and has the feature of allowing attributes of internally stored keys to change, resulting in global mutable state in the model. It is well known that a naïve implementation of the API is insecure - the use of attributes must be constrained. We have found several variations of attacks and validated some new fixes. Results appear in Tsalapati's MSc thesis, [16], and work is ongoing in collaboration with Armando's group in Genova. In particular, we are working towards an analysis of the fixes proposed by Eracom in their ProtectServer product.
- (iv) **EMV** In his MS thesis, Alvanopoulos did some exploratory work on API analysis for point of sale terminals implementing the EMV 'Chip and PIN' standard, using first-order theorem proving, [1]. Although his results did not include any previously unknown flaws, his thesis suggests several areas for further investigation.

2.3 Achievement of Objectives

We reproduce below *literatim* the original objectives of the project (in italics), followed by a brief evaluation of the extent to which they were achieved or surpassed.

1. *Devise a technique for reasoning about cryptographic protocols such that the complexity of breaking certain cryptographic functions, and the effect on this complexity of information extracted from previous operations, can be reasoned about in an automated framework.*

Advances 2, 3, and 4 addressed this objective. There is still work to be done, however: for example, to fully develop the cryptographic foundations of the quantitative reasoning used in the Parallel Key Search model.

2. *Develop a formalism similar to those currently used for cryptographic protocol analysis suitable for analysing the protocols of electronic payment systems.*

All the advances address this objective. The successful case studies show that we have made considerable progress towards this.

3. *Extend the CORAL tool for cryptographic protocol analysis to use this technique and formalism, producing an effective and usable prototype tool for analysis of security APIs.*

We chose to extend other existing tools and develop new ones rather than extend CORAL. Again, the successful case studies show that our new tools are effective and usable.

4. *Improve electronic payment system security and prove the effectiveness of the tool by discovering new attacks and security results on electronic payment systems, and together with collaborators, finding fixes for the attacks.*

As shown in the case studies above, we discovered a number of previously unknown flaws, and proposed and verified several fixes.

5. *Produce an accurate evaluation of the new tool.*

The tools we produced were evaluated in our own work (e.g. [12, 4]) and also in the MSc theses of students who used and improved the tools, [1, 6, 16].

3 Project Plan Review (Research Planning and Practice)

The vast majority of the work in the project was carried out by the RA, Dr. Graham Steel, who designed the project, recruited the collaborators, authored the original proposal, and directed the day to day running of the project. Prof. Bundy acted as a mentor. Dr. Fleuriot was largely unavailable during the project period due to sabbaticals and fellowships.

At the start of the project, during a visit to the University of Cambridge security lab (where most known API attacks had been manually discovered), it quickly became clear that there were two very distinct classes of API level attack: those where a critical piece of data is revealed by some combination of API commands and brute force guessing, and those where critical data (usually a PIN) is never actually revealed, but the intruder can work out its value from the error messages received from the device (a ‘covert channel’ attack). These two different types of attack required different analysis techniques. For the former, we had to deal with normal ‘Dolev-Yao’ style operations, algebraic properties of cryptofunctions such as XOR, and brute guessing steps such as key conjuring and parallel key search. A model for these was developed incrementally throughout the project, combining theoretical work with practical experiments. Early results presented at CADE in 2005 [10] attracted the interest of the security group at INRIA Nancy, and led to Dr. Steel making a visit there in early 2006. A fruitful and ongoing collaboration developed.

In 2005, details emerged of a joint project between MIT and Cambridge, also aimed at applying formal tools to the problem of API analysis. This project was only moderately successful. The main thrust was to use the Otter theorem prover to rediscover known attacks on the IBM 4758 CCA API. In summer 2006, an MSc student (Gavin Keighren) joined our project team with the brief of using API analysis models developed during the first visit to Nancy and the protocol analysis tool CL-AtSe (which supports XOR natively) to analyse not just the old version of the CCA API with the known attacks, but the new version released by IBM to fix the vulnerabilities. Keighren was able to rediscover the known attacks much faster than the Otter-based approach, and in a more realistic model with more commands modelled. He also discovered a flaw in the supposedly fixed version, which was reported

to IBM, [7]. In later work, we proposed and verified a patch for the flaw using an implementation of a decision procedure arising from theoretical work in collaboration with the Nancy team, [4].

A visit to our collaborators nCipher in early 2005 revealed more interest in PIN processing attacks. Dr. Steel developed a Markov decision process model for these attacks, and a way of automatically constructing models for a given API using constraint logic programming. In the summer of 2007, another MSc student, Eirini Kaldeli, made several improvements to this system, resulting in smaller models and faster run-times for analysis, [6].

In January 2006, KAL, a local SME specialising in ATM software, approached Dr. Steel to propose a joint project aimed at security analysis of KAL's runtime environment for ATMs. Funding was obtained from Scottish Enterprise, and Dr. Steel was seconded half-time to the project for 6 months between July 2006 and January 2007. The project was a success, and several improvements to KAL's system were suggested through the use of security protocol and API analysis techniques, [14].

ITI techmedia, an investment arm of Scottish Enterprise, have expressed interest in commercialising some of the API analysis techniques developed in the project. A tender for a feasibility study was submitted to the ITI in August 2007, for work to take place in the second quarter of 2008.

After the MIT-Cambridge project ended, one of the participants (Jon Herzog) wrote an article in the IEEE Security and Privacy magazine arguing that modelling of global state was a major problem for API analysis, citing the PKCS#11 API as an example. To investigate this, a further MSc student, Eleni Tsalapati, experimented with models of the PKCS#11 API, and drew similar conclusions, [16]. In recent work in collaboration with Armando's group in Genova, we have been able to build on the Tsalapati work to produce accurate (in terms of global state modelling) models that can be analysed in reasonable time. This work is ongoing, with the aim of validating or attacking the fixes for PKCS#11 proposed by Eracom, and used in their ProtectServer product.

In late 2006, it seemed that the time was ripe for a meeting of researchers in the area of API analysis. Dr. Steel recruited an organising committee consisting of Jon Herzog (Naval Postgraduate School) and Mike Bond (Cryptomathic), and set about organising the first international workshop on Analysis of Security APIs, which took place as a satellite of the IEEE Computer Security Foundations symposium in July 2007. Project partners nCipher provided sponsorship, as did Cryptomathic. The workshop attracted 16 participants who saw 8 high quality presentations, and was generally agreed to have been a great success. A second API analysis workshop is already planned for 2008, and Andy Gordon (Microsoft Research) wrote about it on his blog as a 'hot area' of research¹. Analysis of security APIs can be said to a healthy and exciting new field of research, due in no small part to the work in this project.

4 Research Impact and Benefits to Society

The high impact of our research is evidenced by the attraction of international collaborators (Cortier - INRIA Nancy, Delaune - ENS Cachan, Courant - Verimag Grenoble, Armando - University of Genova), the founding of an international workshop series with the assistance of the other main players in the field (Bond at Cryptomathic and Herzog at the Naval Postgraduate School), and the expressions of interest from industry (nCipher were prepared to pay for our first workshop, KAL hired Dr. Steel to analyse their systems, and Utimaco have expressed interest in a future collaborative project). Furthermore, ITI Techmedia are considering the direct funding of commercialisation of the research as part of their Software Integrity Engineering programme (our tender is currently under consideration). Our discoveries of flaws and proposed fixes have already improved the quality of products deployed 'in the field', and with the possibility of investment to bring our tools up to a commercial standard, there is potential for an even greater impact in the future.

5 Further Research and Dissemination Activities

In the immediate future, Dr. Steel will spend a year working as a researcher at ENS-Cachan, pursuing further research in API analysis, before returning to Edinburgh to take up a permanent post as Lec-

¹<http://whigmaleerie.spaces.live.com/blog/cns!C6149B019D236BF5!340.entry>

turer in Computer Science. He is also preparing an application for an EPSRC Leadership Fellowship. One of the MSc students who worked on the project, Gavin Keighren, will commence a PhD at the University of Edinburgh in October 2007, with security API analysis as his research topic.

5.1 Output of Research Staff

As recognition of his role as an independent researcher, Dr. Steel was promoted to AR2 (now grade 8) in August 2006. He was awarded an open ended contract at the University of Edinburgh in April 2007. He will be counted as category ‘A’ staff for the 2007 RAE, and has recently been appointed as a Lecturer starting from 1st October 2007.

Dr. Steel supervised four MSc students who carried out projects related to the grant. One of them, Gavin Keighren, won the best student award in 2006. Three are now taking up PhD positions, and one is continuing to seek funding for a position.

5.2 Communication of Research

In the original case for support, we set a target of two papers in major international conferences and one journal paper. This target was surpassed. Four conference papers [10, 13, 4, 2] and one journal paper [12] have already been published, along with three workshop papers [5, 11, 15]. A further journal paper is under review [8], and a third is in preparation, [3].

Throughout the project, Dr. Steel gave seminars on the project to top academic and industrial research groups, including the University of Cambridge Computing Laboratory (Oct 2004), Motorola (Aug 2005), Toyota (May 2005), the University of Genova (Sept 2005 & Sep 2007), ETH Zürich (Oct 2005), INRIA Nancy (Jan 2006 & Sept 2006), the University of Birmingham (Feb 2006), the University of Glasgow (April 2006), Verimag Grenoble (Sept 2006), ITU Copenhagen (Nov 2006), Siemens Corporate Technology (Munich, Dec 2006), Ruhr University Bochum (April 2007) and Utimaco (Aachen, April 2007). Dr. Steel also presented the work at international conferences including CADE (July 2005), TACAS (April 2007) and CSF (July 2007), and also at workshops, including Scottish Theorem Provers (Oct 2004 & March 2006), UITP (April 2005), CIAO (Apr 2005, Apr 2006 & Apr 2007), Automated reasoning workshop (Apr 2005), CSFW (July 2006), FCS-ARSPA (Aug 2006), FCC (July 2007), and the first Analysis of Security APIs workshop (July 2007). He also made research visits to nCipher (Jan 2005), and CESG (GCHQ, May 2005 & April 2006). Prof. Bundy gave a talk on the project to a visiting delegation from Microsoft (May 2005). Visiting researchers in the field were hosted in Edinburgh, including Mike Bond and Jolyon Clulow (April 2005 and Sept 2005) and Judicaël Courant (June 2006).

Details of the project, including publications, were disseminated online from the project webpage² and also via a case study on the PRISM webpages³. Programs were made available for download⁴. Theorem proving problems in first-order logic arising from the work were included in the TPTP library (version 3.2.0 onwards), and were used in the 2006 theorem proving competition (CASC-J3).

Dr. Steel gave guest lectures on the project to undergraduates and postgraduates at the University of Edinburgh (Oct 2004, Nov 2005 & Nov 2006) and at the University of St. Andrews (March 2005).

To communicate the research to the wider public, Dr. Steel employed a variety of channels, including a poster at the Edinburgh Sci-Fun Festival (April 2006), an interview for UK Future TV⁵, and an entry into the EPSRC Computer Science Writing Competition 2007, which reached the shortlist but failed to win a prize, and is available from the project webpage. An interactive ‘PIN Cracking’ demonstration is being constructed to demonstrate the science behind the project to school children at University open days and science fairs. A poster about the project was also presented at the EPSRC ICT review day in Edinburgh (Dec 2006).

Details of the spin-off project with KAL appeared in Infinity (a University of Edinburgh in-house magazine) and on the Scottish Executive webpages.

²<http://dream.inf.ed.ac.uk/projects/aascs>

³<http://www.cs.bham.ac.uk/~dxp/prism/casestudies/pin cracking.php>

⁴<http://homepages.inf.ed.ac.uk/gsteel/CCA-experiments/files/verifier/http://homepages.inf.ed.ac.uk/gsteel/anablock/>

⁵<http://www.ukfuturetv.com/grahamsteel.wmv>

6 Explanation of Expenditure (Cost Effectiveness)

The largest deviation from planned expenditure was that we did not employ a CO to code up a syntax translator, as budgeted for in the original proposal. There were two reasons for this: the first was that as we incrementally developed the ability of our tools to cover new kinds of API attack, the input language evolved, and is still in a state of flux, reducing the utility of a translator. The second is that a reorganisation of personnel at the University meant that our original candidate for the job was unavailable.

As can be seen in section 5.2, the travel budget was stretched to a remarkable number of conference, seminar and research visits, using cost-saving measures such as low-cost airlines, overnight trains, and accommodation in friend's spare rooms. There was a slight underspend on the budget for purchase of standards, which was in turn used for further research visits. Contributions in kind were made by nCipher and CESG. nCipher additionally contributed £1000 to the cost of the Analysis of Security APIs workshop, and Cryptomathic contributed £200.

Publications Arising from the Project

- [1] I. Alvanopoulos. Security API analysis of an EMV TPM-enabled system. Master's thesis, University of Edinburgh, 2007.
- [2] V. Cortier, S. Delaune, and G. Steel. A formal theory of key conjuring. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF20)*, pages 79–93, Venice, Italy, 2007.
- [3] V. Cortier, S. Delaune, and G. Steel. A formal theory of key conjuring. *Journal of Computer Security*, 2007. In preparation.
- [4] V. Cortier, G. Keighren, and G. Steel. Automatic analysis of the security of XOR-based key management schemes. In O. Grumberg and M. Huth, editors, *TACAS 2007*, number 4424 in LNCS, pages 538–552, 2007.
- [5] V. Cortier and G. Steel. On the decidability of a class of XOR-based key-management APIs. In *Proceedings of the FCS-ARSPA workshop*, August 2006.
- [6] E. Kaldeli. Investigating formal representations of PIN block attacks. Master's thesis, University of Edinburgh, 2007.
- [7] G. Keighren. Model checking security APIs. Master's thesis, University of Edinburgh, 2007.
- [8] G. Keighren, G. Steel, and A. Bundy. Checking security APIs with protocol analysis tools. Submitted to ACM Transactions on Information and System Security.
- [9] G. Steel. Security report: nCipher payshield API. Sent to nCipher in February 2006.
- [10] G. Steel. Deduction with XOR constraints in security API modelling. In R. Nieuwenhuis, editor, *Proceedings of the 20th Conference on Automated Deduction (CADE 20)*, number 3632 in Lecture Notes in Artificial Intelligence, pages 322–336, Tallinn, Estonia, July 2005. Springer-Verlag Heidelberg.
- [11] G. Steel. Visualising first-order proof search. In *Workshop on User Interfaces for Theorem Provers (UITP '05)*, pages 179–189, Edinburgh, Scotland, April 2005.
- [12] G. Steel. Formal analysis of PIN block attacks. *Theoretical Computer Science*, 367(1-2):257–270, November 2006. Special Issue on Automated Reasoning for Security Protocol Analysis.
- [13] G. Steel. The importance of non-theorems and counterexamples in program verification. In *Position Papers of the ETH-Zürich VSTTE conference – Verified Software: Theories, Tools, Experiments*. Springer, 2006. To Appear.
- [14] G. Steel. Recommendations. Confidential report delivered to KAL, January 2007.
- [15] G. Steel and J. Courant. A formal model for detecting parallel key search attacks. In *Proceedings of the 3rd workshop on Formal and Computational Cryptography (FCC '07)*, 2007. Informal proceedings to appear as an INRIA research note.
- [16] E. Tsalapati. Analysis of PKCS#11 using AVISPA tools. Master's thesis, University of Edinburgh, 2007.