

Formal Verification for Survivable Key Management Systems

Irfan Zakiuddin, DERA Malvern
I.Zakiuddin@eris.dera.gov.uk

Jim Woodcock, Oxford University Computing Lab and Formal Systems (Europe) Ltd
jim.woodcock@comlab.ox.ac.uk

Michael Goldsmith, Formal Systems (Europe) Ltd
michael@fsel.com

Jason Hulance, Formal Systems (Europe) Ltd
jason@fsel.com

1 Introduction

Key management systems are central to secure informations systems. Changing paradigms in technology and the increasing dependency on technology, by all aspect of society, will mean that key management systems will be more important than ever. In particular, future key management systems must be :

1. highly dependable, capable of surviving a range of attack;
2. scalable, to very large numbers of nodes and very, very large numbers of keys;
3. dynamic, providing services across networks of variable connectivity, and changing sets of principals.

Studies conducted by DERA have also identified that future key management systems should also:

4. merge key management functions onto backbone networks, rather than having separate, dedicated key management networks;
5. utilise the civil communications infrastructure, for key distribution, where necessary.

Together, the five requirements above, raise very significant questions for high integrity design, implementation and accreditation of survivable key management systems.

In recent months, as part of DERA's formal verification program, we have looked at applying formal verification techniques to support development and accreditation of such highly dependable and complex systems. This position statement discusses challenges and progress.

2 Formal Verification

The discipline of formal verification has a long history, much of which is dominated by automated proof tools. These tools were cumbersome and their practical applications were limited. The field was revolutionised in the mid-90s by model-checking technologies. A model-checker performs an exhaustive search over all the behaviours of a model, seeking satisfaction of a given

property. Model-checkers have the advantage of being:

- * relatively quick and easy to use, compared to proof tools,
- * able to find 'bugs', as a result of their exhaustive search.

The exhaustive search performed by a model-checker is particularly useful, for two reasons:

- * the 'bugs' it finds are sometimes very subtle, so hard to detect by other means,
- * the behaviour leading to the bug is also shown by the tool and this invariably sheds light on design flaws.

Industrial applications of model-checking have, thus far, been dominated by the hardware industry. But from 1995 to 1997, work at DERA, Formal Systems (Europe) Limited and the Oxford University Computing Lab made striking advances applying the FDR model-checker to verifying authentication and authenticated key agreement protocols. As part of that work a protocol thought, for 17 years, to be sound and 'proven' to be so, was shown to be flawed [1]. Since then DERA has developed an active program applying FDR to a range of mission critical system including:

- * integrating land attack missiles into Royal Navy submarines,
- * avionics architectures,
- * tactical internets,
- * the MAFTIA architecture.

More generally, the formal verification community is thriving, with powerful new ideas constantly emerging [2,3]. Model-checking dominates the community but exciting ideas to combine verification techniques are emerging [4].

Thus formal verification is a dynamic field, with a substantial track record and tools and techniques of increasing power. It is natural to seek solutions for designing and assessing the dependability of high assurance, survivable key management systems based on formal verification tools.

3 Current Investigations

Current work applying formal verification techniques to key management problems has two principle thrusts:

1. Verifying survivable key hierarchies.
2. Verifying attack tolerant, secure multicast protocols.

The driving philosophy behind both items is that to achieve an architecture which is both survivable and scalable, it is essential to consider both these critical attributes together. In short, designing a system with only survivability in mind and then hoping it will scale is a flawed philosophy.

Both items are recently initiated work in progress, they are briefly reviewed here.

3.1 Verifying Survivable Key Hierarchies

Systems for scaling key management can use techniques analogous to those used to scale network management. In particular, the 'a WAN-of-LANs' paradigm is useful (the term is due to Paulo Verissimo of the University of Lisbon, but the concept is common). This means that the wide area network is decomposed into small local area networks. These local area networks can form secure key management domains; each is delegated some responsibility for key management. Communication between domains is mediated by the domain gateway, and the wide area network results from the domains and their inter-domain links. Discussions of the benefits of the approach, for key management can be found in [5,6] (and others), but in summary the benefits are:

- Scaling key management functions,
- Delegated security management,
- Enabling interoperability,
- Simplified partition management,
- Better breach containment.

Our work is using formal verification techniques to validate the last two of the listed benefits. Partition management, is identify and to respond to loss of connectivity. Breach containment means to localise a security breach to as small a domain as possible. These are both survivability properties.

The techniques we are using were developed to verify models of ad-hoc networks. Highly mobile environments require self forming and self-managing networks, these are ad-hoc networks [7]. Some of these networks are hierarchical, and in other work we developed formal verification techniques to analyse ad-hoc networks, especially hierarchical ones. The techniques developed there are very relevant for verifying partition management and breach containment properties of mobile key management hierarchies. Our research is currently concentrating on adding a hierarchy to models of the Cliques work [8] and verifying partition management and breach containment properties.

We are also applying formal verification to study key hierarchies from the work of Birman, Dolev and Rodeh [9]. This work presents protocols for group re-key which tolerate network partitions and node failures. The aim of the protocols is to create and to maintain a hierarchy of keys.

3.2 Verifying Attack Tolerant, Secure Multicast Protocols

The second item (led by Jim) is to verify 'A High Throughput, Secure Reliable Key Management Protocol', developed by Reiter and Malkhi [10], this type of protocol is used in Omega key management system, from AT&T Labs [11]. This protocol is interesting from a key management systems perspective because:

- it combines good performance with attack tolerance,
- it may be implemented over a public communications infrastructure, like the internet.

But the protocol is also interesting from a verification perspective. To achieve its performance the protocol's nodes maintain a complex graph of message acknowledgements. Modelling complex datatypes is a challenge for mainstream model-checking tools; they are designed to explore a space

of behaviours arising from the interactions of small local state processes.

Our planned solution was to combine the Alloy Constraint Analyser (ACA) tool [12], with FDR. This tool is probably unique amongst model-checkers in that it verifies properties of datatypes, and relations defined over datatypes. The plan was to use ACA to verify node state and FDR to verify node interaction. Unfortunately, ACA was not able to scale to verify properties of a nodes state. Nevertheless, use was made of ACA to produce correct formalisations of the protocols properties and the Z/Eves theorem prover was used to verify an abstraction of the Malkhi-Reiter protocol against these properties.

4 Conclusions and Prospects

These items are very much work in progress and we hope to have results by late October. Two important conclusions emerge from our work. Firstly, formal verification is applicable to increase assurance, in particular, in survivable key management systems, and, in general, in high assurance networks. Secondly, from a formal verification perspective, the work based on the Malkhi-Reiter protocol is significant. It is common for group communication protocols to combine complex process state with complex process interaction. Thus, the limitations discovered in that item are likely to apply for verifying other dependable group communication protocols. A practical way ahead was found, but this required a manual proof assistant. If a more automated solution is required, then tools (such as ACA) and techniques (such as combining ACA with FDR) will need to mature. We can conclude that this item informs research directions for developing formal verification tools and techniques to assess dependable group communication systems.

Both the items discussed here are innovative applications of tools and techniques, distinctive from most of the work in the verification community.

In future, we also hope to apply techniques developed here to study protocols for distributing the functions of a key management node, this is done to give survivable service availability, without replicating confidential data. We also hope to plan and execute a systematic approach to verifying key management systems, incorporating:

- requirements analysis,
- vulnerabilities analysis,
- case studies of survivable key management integrated onto backbone networks,
- structured approach to formal verification of the whole system.

5 References

1. G. Lowe. Breaking and Fixing the Needham Schroeder Public Key Protocol Using CSP and FDR. In proceeding of TACAS '96, LNCS 1055
2. E Emerson (editor). CAV 2000, Proceeding of the 12th International Conference on Computer Aided Verification. LNCS 1855
3. R Lazić. A Unifying Approach to Data-Independence. Oxford University Computing Laboratory Technical Report TR-4-00.
4. N Shankar. Combining Theorem Proving and Model-checking through Symbolic Analysis. Invited paper at CONCUR 00.
5. <http://www.glomo.sri.com/aswin/>
6. <http://www.ietf.org/internet-drafts/draft-ietf-ipsec-gkmframework-02.txt>
7. http://tonnant.itd.nrl.navy.mil/manet/manet_home.html
8. <http://www.isi.edu/~gts/CLIQUES/>
9. O. Rodeh, K. Birman and D. Dolev. Optimized Group Rekey for Group Communication Systems. Network and Distributed System Security 2000, February, San Diego, California. Available from: <http://www.cs.huji.ac.il/~orodeh/>
10. D. Malkhi and M. Reiter. A High-throughput Secure Reliable Multicast Protocol. Journal of Computer Security, IOS Press, 1997.
11. M. K. Reiter, M. K. Franklin, J. B. Lacy, and R. N. Wright. The Omega Key Management Service. Journal of Computer Security 4(4):267-287, IOS Press, 1996.
12. <http://sdg.lcs.mit.edu/alloy/>