Performance modelling of security protocols

Nigel Thomas and Yishi Zhao

School of Computing Science, Newcastle University, UK.
nigel.thomas@ncl.ac.uk

Abstract. In this paper we state the case for the performance evaluation of secure systems. We provide evidence of a wide variation in performance of different secure methods. We further explore the overhead introduced by secure functions in considering a case study in non-repudiation. We present a model of a non-repudiation protocol specified using the Markovian process algebra PEPA and present results derived using mean value analysis and mean field approximation.

1 Introduction

The security of modern computer and communication systems is a major concern for governments, organisations and individuals, resulting in a significant effort to ensure, and prove, that systems remain secure and data remains private. However, it is also essential that security measures do not impose excessive constraints on the user which then encourage subversion of those measures in order to make the system more usable. It should be clear that any security measure that degrades usability is undesirable. However, all security measures will entail some additional work being undertaken which imposes a performance overhead. It is therefore essential that this overhead is understood, measured and minimised. In some practical situations there may be a choice of methods (including varying protocols, algorithms or parameters) which could be employed. Changing the choice of method could have a potentially significant impact on the system performance without degrading the security. In other situations methods can be modified (e.g. by changing a key length or refresh rate) which might improve performance at the cost of some level of security, thus giving a security performance trade-off [1]. However, quantifying this trade-off is not generally possible due to a lack of quantitative methods for evaluating system security. Instead our approach is based purely on evaluating the performance of the system and thus giving the system designed information on competing designs.

Cryptographic protocols are one of the few areas of security have been received attention from both security and performance communities [2–5]. For example consider Figures 1-3, which illustrate the performance of different aspects of secure system behaviour. Figure 1 shows the average execution time for a variety of cryptographic protocols for a specific message length. Figure 2 illustrates the dramatic variation in performance that can be achieved by varying the key length for public key encryption algorithms. Figure 3 shows the measured performance of different functions within a secure stock trading application, with encryption and without. By considering data such as this, a system designer can modify a secure system to take account of performance.

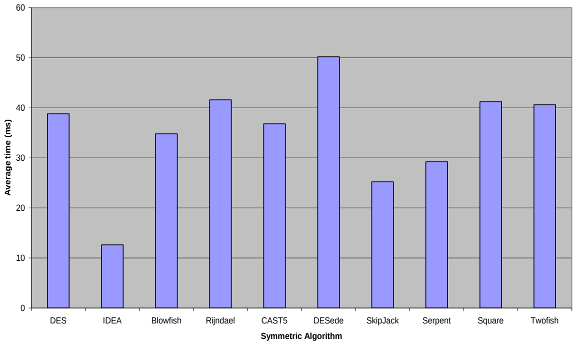

Fig. 1. Average execution time for symmetric key encryption algorithms [4]

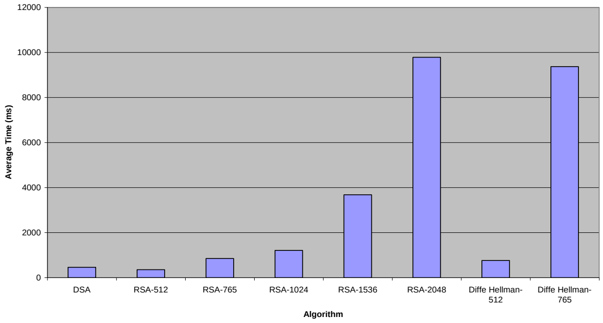

Fig. 2. Execution time varied against key length for RSA and Diffie-Hellman [2]

2 Performance models of secure systems

A greater level of understanding of secure system performance can be gained by specifying and analysing a performance model. A *Key Distribution Centre* (key exchange protocol) has been studied in our previous work, which shows the possibility of modelling by a stochastic process algebra PEPA and analysis by several alternative techniques [7–9]. The advantage of using a formal specification for such models is that it is possible to check specific properties to ensure that the model correctly depicts behaviour which is essential to the security of the

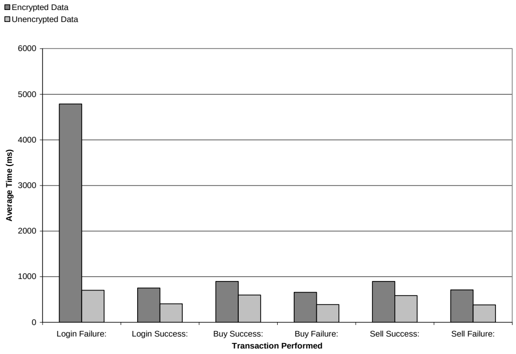

Fig. 3. Average response time for functions within a secure trading application [6]

system. Thus a formal performance model and a formal security model of a given system can be shown to exhibit equivalence, giving the system designed some reassurance that the performance behaviour is valid. A process algebra allows detailed behaviour to be modelled and has the potential to be modified automatically through model transformations to facilitate alternative forms of analysis.

3 A case study in non-repudiation

A non-repudiation service will prevent either of the principals involved from denying the contract after the agreement. The two such protocols were proposed by Zhou and Gollmann [10, 11] and use a non-repudiation server, known as a *Trusted Third Party* (TTP). We will specify the first of these protocols, from now on referred to as ZG1.

- A: originator of the non-repudiation exchange
- B: recipient of the non-repudiation exchange
- TTP: on-line trusted third party provide network services accessible to the public
- M: message sent from A to B
- C: ciphertext for message M
- K: message key defined by A
- $NRO = sS_A(f_{NRO}, B, L, C)$: Non-repudiation of origin for M
- $NRR = sS_B(f_{NRR}, A, L, C)$: Non-repudiation of receipt of M
- $sub\_K = sS_A(f_{SUB}, B, L, K)$: proof of submission of K
- $con\_K = sS_T(f_{CON}, A, B, L, K)$: confirmation of K issued by TTP

First, A sends the ciphertext (C) and a non-repudiation origin (NRO) for message M to B, and then B replies back with a non-repudiation receipt (NRR) to A. Now B possesses the ciphertext, but cannot read it as he still hasn't got the key to decrypt M. According to the non-repudiation requirement, B is not a trusted agency to A for sending the key directly to B, they only can resort to a trusted third party (TTP). After receiving the key and proof of submission (sub_K), the TTP will generate a confirmation of K (con_K) and publish in a read only public area. Finally, B can get the key from this public area to decrypt ciphertext (C) and A fetches the confirmation of submission as non-repudiation evidence.

From this protocol specification we can derive the following PEPA model for the complete system when there are N pairs of principals.

$$\begin{aligned}
TTP &\overset{def}{=} (publish, r_p).TTP \\
AB_0 &\overset{def}{=} (sendB, r_b).AB_1 \\
AB_1 &\overset{def}{=} (sendA, r_a).AB_2 \\
AB_2 &\overset{def}{=} (sendTTP, r_t).AB_3 \\
AB_3 &\overset{def}{=} (publish, r_p).AB_4 \\
AB_4 &\overset{def}{=} (getByA, r_{ga}).AB_5 \\
&\quad +(getByB, r_{gb}).AB_6 \\
AB_5 &\overset{def}{=} (getByB, r_{gb}).AB_7 \\
AB_6 &\overset{def}{=} (getByA, r_{ga}).AB_7 \\
AB_7 &\overset{def}{=} (work, r_w).AB_0 \\
SystemZG1 &\overset{def}{=} TTP[K] \underset{publish}{\bowtie} AB_0[N]
\end{aligned}$$

AB_0 to AB_7 in the above ZG1 PEPA model denote the different behaviours of the AB component, and its evolution along the sequence of prescribed actions in the protocol. The choice from AB_4 to AB_5 and AB_6 means step 4 and step 5 in ZG1 can happen in any order. The *work* action is used to define that B can do something with the key and ciphertext after he has obtained these, before returning to the state AB_0 to make a new request again, which forms a working cycle to investigate the steady state.

Figure 4 shows the average queue length varied with number of customer involved in this non-repudiation system solved by an ODE (approximate) solution supported by the PEPA tools [**?**] and by exact mean value analysis [**?**].

The ODE approximation does not depend on deriving the state space of the underlying Markov chain, hence it scales very well [**?**]. Furthermore, the solution converges to the exact solution as the number of components increases, thus it becomes extremely attractive for solving models of extremely large systems, as illustrated in Figure 5.

Finally we show the relative performance between the ZG1 protocol and another proposed by the same authors which is shown to be less scalable. Thus a designer could use such a comparison to choose the appropraite protocol.

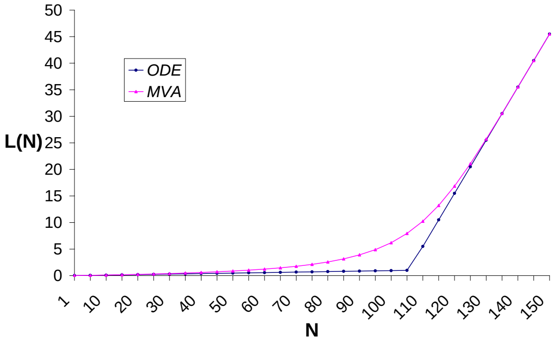

Fig. 4. Average queue length varied with population size calculated by the ODE, $r_b = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_w = 0.01, K = 1$

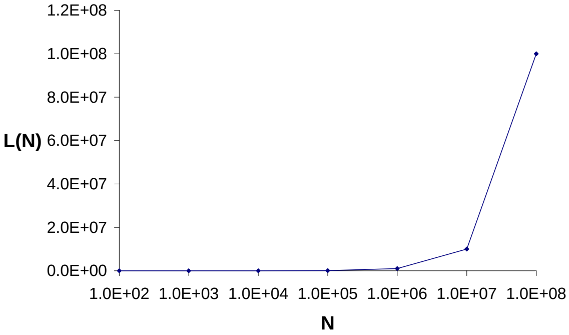

Fig. 5. Average response time varied with population size calculated by the ODE, $r_b = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_w = 0.01, K = 1$

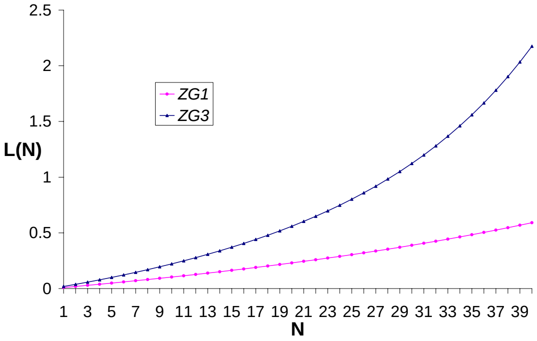

Fig. 6. Average number of waiting jobs with ZG1 and ZG3, $r_p = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_w = 0.01$

4 Conclusions

In this paper we have explored the issue of modelling secure systems. Estimating the costs (as well as the benefits) of security is important. We have shown that not only can there be an appreciable overhead introduced by secure methods, but also that the overhead can vary considerably according to the method employed. Thus there is a clear opportunity for the system analyst to improve, or even optimise, performance by choosing or tuning the various algorithms and protocols.

To date our analysis has focussed on identifying and employing efficient solution methods. There is considerable scope for further work to investigate the relationship between formal security models and formal performance models. The ultimate goal would be to create a system which could automatically produce analysable performance models from security models. However, the choice of security solution, driven by the performance security trade-off should always remain an expert task.

References

1. K. Wolter, and P. Reinecke. Performance and security tradeoff, in: *Formal methods for quantitative aspects of programming languages*, pp. 135-167, Springer, 2010.
2. S. Dick and N. Thomas, Performance analysis of PGP, in: F. Ball (ed.) *Proceedings of 22nd UK Performance Engineering Workshop*, Bournemouth University, 2006.
3. W. Freeman and E. Miller, An Experimental Analysis of Cryptographic Overhead in Performance-critical Systems, *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*,IEEE Computer Society, 1999.
4. C. Lamprecht, A. van Moorsel, P. Tomlinson and N. Thomas, Investigating the efficiency of cryptographic algorithms in online transactions, *International Journal of Simulation: Systems, Science & Technology*, 7(2), pp 63-75, 2006.
5. M. Buchholtz, S. Gilmore, J. Hillston and F. Nielson, Securing statically-verified communications protocols against timing attacks, *Electronic Notes in Theoretical Computer Science*, 128(4), Elsevier, 2005.
6. L. Thorpe, unpublished report, University of Durham.
7. Y. Zhao and N. Thomas, Approximate solution of a PEPA model of a key distribution centre, in: *Performance Evaluation - Metrics, Models and Benchmarks: SPEC International Performance Evaluation Workshop*, pp. 44-57, LNCS 5119, Springer-Verlag, 2008.
8. N. Thomas and Y. Zhao, Fluid flow analysis of a model of a secure key distribution centre,*Proceedings 24th Annual UK Performance Engineering Workshop*, Imperial College, 2008.
9. Y. Zhao, N. Thomas, Efficient solutions of a PEPA model of a key distribution centre, *Performance Evaluation*, **67**(8), pp. 740-756, 2010.
10. J. Zhou and D. Gollmann, A Fair Non-repudiation Protocol, in: *Proceedings of IEEE Symposium on Security and Privacy(SP'96)*, IEEE Computer Society, 1996.
11. J. Zhou and D. Gollmann, Observation on Non-repudiation, in: *Advances in Cryptology-ASIACRYPT'96*, 133-144, LNCS 1163/1996, Springer-Verlag, 1996.