Jikzi – A New Framework for Security Policy, Trusted Publishing and Electronic Commerce

Ross J. Anderson

Cambridge University Computer Laboratory
Pembroke Street
Cambridge CB2 3QG
England
rja14@cl.cam.ac.uk

Jong-Hyeon Lee

Filonet Korea Inc.
CBS Building, 917-1 Mok-dong
Yangchun-gu, Seoul 158-701
Korea
jhlee@filonet.com

Abstract

In this paper, we describe a thread of research which we have followed off and on at Cambridge for about three years. Our topic is the security of electronic documents, in the broad sense: how can we be sure of the authenticity of things that are published electronically?

This started off as a relatively small project, which we thought would take only a few weeks. The goal was to help our medical informatics department publish information such as drug formularies and treatment protocols on the hospital LAN or PC diskettes in an appropriately dependable way. It rapidly became clear that the problem was much larger and more complex; a general solution would not only cope with 'content' – text, audio, video, software, whatever – but also with objects such as public key certificates. If done properly, it would give us a systematic way to deal with security policy on the web.

Our goal now is to let people build integrated publishing and e-commerce services using simple, uniform and appropriate mechanisms. Our proposed solution is a single transparent markup language that allows us to support multiple security policies, plus supporting material ranging from a test implementation to an authentication logic.

1 Introduction

In 1996, we were approached by a team developing a medical hypertext system at our local hospital. UK doctors increasingly have a PC in front of them as they consult their patients and use it to store the patient's medical record; so it was felt that the time was ripe to transfer into electronic form all the reference books used routinely during consultations. Funding had been secured from the National Health Service, and a first prototype built. The designers now felt that they needed to secure the preproduction and later versions, mainly against errors but also against attacks, and asked us for advice. We thought initially that this would be a simple project, lasting only a few weeks, in which we would show them how to implement a digital signature system based on X.509. How wrong we were!

The system, which is called Wax, challenged our thinking about document security in a number of ways. The naive solution, namely to have the publisher sign each book in the system, turned out to be too expensive; it is not efficient to verify the signature on a whole book when the typical reader simply wishes to consult a paragraph relating to a particular drug or disease. There was also a problem with key management: the X.509 system was originally designed for electronic phone books and has since been adapted for financial transactions, so its model is a key lifetime of 2–3 years and a signature lifetime of 2–3 days. However, an electronic book will typically contain long-lived data, and to protect it using relatively short-lived objects such as public keys is problematic. How, for example, would one handle revocation? One can always use a hash-based timestamping service to certify that a signature was received in advance of a revocation certificate cancelling the key which certified it, but in that case why not protect the book using the hashing mechanism rather than the digital signature?

Considerations like these led us to design Wax so that each publisher has associated with it a tree of hashes: the leaves are hashes of the individual sections, and the nodes progress up through chapters and books until the root, which protects a publisher's whole catalogue, is authenticated by other means at system startup. (In the initial version of

Wax, this was a standard RSA signature, but the costs of licensing the product for export to the USA led us to use a one-time signature instead for later versions.) Thus when a user opens a book at the entry corresponding to a particular drug, the entry can be verified quickly without having to check a signature on the entire book. Wax therefore taught us to use hash trees and minimise the amount of signature – preferably to a single signature on each new version of a publisher's catalogue.

Wax is described in detail in [1], and has had some success among UK and US medics. But it uses a proprietary hypertext format, so when we were next asked to look at an online medical application (the British National Formulary, which is the list of all drugs approved for prescription in the UK [2]) the obvious step was to generalise our ideas to work with html. The result was the ERL, or eternal resource locator, which is described in [3]. An ERL is essentially a URL plus a hash of what one should expect to find there. Thus a medical publisher can modify his online catalogue to replace the URLs of his online books with ERLs, and now users whose browers have a plugin which can parse and verify the hashes, get the same protection as in Wax; meanwhile, users with a standard browser enjoy backwards compatibility in that the ERLs function perfectly well as ordinary URLs.

In order to cope with web pages that change frequently, we added the facility for an ERL to contain the hash of a public key with which a digital signature of the indicated content could be verified. Such keys do not incur the overhead of a traditional public key infrastructure but are rather seen as 'flexible links' which can be inserted in an otherwise static trust structure wherever they are needed. In this way, a book of largely static information such as drug data can also incorporate some dynamic content such as links to the latest safety bulletins.

Having developed this, we realised that we now had a surprisingly powerful and general protection system. The writer of a web page or other document can vouch for any digital object by simply including its ERL, and this enables the construction of webs of trust of the kind familiar from PGP but applying to quite general objects rather than just to the names of principals. In applications where trust structures are relatively static, many of the problems associated with public key infrastructures simply go away; one ends up managing a few root keys and doing a lot of version control.

By late 1998, it had become clear that rather than adding proprietary extensions to html, we ought to support XML as the emerging framework for commercial markup languages [4]. XML is well positioned between SGML and HTML; it is easier to use than SGML, yet provides more degrees of freedom in extension than HTML. Major software vendors are committed to supporting it: both major players in the browser market already support XML processing partially and are expected to support it fully in the near future.

By this time, too, a number of governments and other organisations had got hold of the idea that signing electronic documents could be a good thing, and further problems had started to emerge as people scrutinised various draft bills on digital signatures. For example, creating a rebuttable presumption of validity for electronic signatures could have the effect of shifting the burden of proof in disputes from the individual to a bank or credit card company, and thus undermine decades of progress in consumer protection legislation [5, 6].

Another consideration was that by then there were several other groups doing work on security markup languages, including IBM's Secure Document Markup Language (SDML) which is designed to support electronic cheques [7], and the World Wide Web Consortium's DSig initiative which enables content rating data to be digitally signed [8]. None of these was general enough to support the kind of functionality needed in our application, so we wanted our next iteration to support a wide range of security policies.

With this in mind, we decided to take a step back and try to place electronic document security in the broader context of computer security research.

2 Context

The first, obvious, point about securing publicly available documents against tampering is the imbalance between research and practice on confidentiality, integrity and availability. Thanks in part to the influence of military funding and the interests of the crypto community, some 90% of research papers deal with confidentiality, 9% with authentication and 1% with availability. But while the military mostly wants to keep data secret, business mostly wants to publish it – most e-commerce is publishing of one form or another (adverts, catalogues, timetables, books, audio, video, software and even public-key certificates). So investment by business is the other way round, with the big money going on backup sites and network redundancy, some money on audit, but only a small amount on encryption[1].

[1]The figures are from the authors' experience of editing 'Computer and Communications Security Reviews'. In the

This bias is particularly noticeable when we look at the security policy models which researchers use to provide a concise and abstract description of the kind of protection which a system requires. The traditional military policy, which we might explain to a layman as *"a clerk with a security clearance to 'secret' can read a file at 'secret' or 'confidential' but not a file at 'top secret' "* was formalised by Bell and LaPadula in two rules: that a process may not read up to a higher level, or write down to a lower one [9]. A very substantial research literature has grown up on top of this, and a number of compliant products have appeared; however, the costs of a rigorous implementation are leading more and more governments to restrict this functionality to specific systems such as firewalls and mail guards.

By the late 1980's, it had already been realised that the requirements of business are different. Clark and Wilson proposed a quite different security policy model, based on the traditional practices of double-entry bookkeeping that had developed in the banking industry since the 12th century; this might be explained to the layman as *"all transactions must preserve an invariant of the system, namely that the books must balance (so a negative entry made on one ledger must be balanced by an equal positive entry on another one); some transactions require two officers to initiate them; and records of payments cannot be destroyed once made."*

A further model, the Chinese Wall model, attempts to describe good practice in businesses such as merchant banking or advertising where a firm may have clients who are competitors. It can be described as *"an executive who has worked recently for one company in a business sector may not have access to the papers of any other company in that sector"*. Yet another, the BMA model, codifies accepted good practice in handling medical records: its executive summary is that *"no-one may see a medical record without the patient's consent, and people with read access may also append information; but no deletions are permitted"*. Details of Clark-Wilson and Chinese Wall may be found in [9], while the BMA model is presented in [10].

This brings us to our second problem, which is that until now there has been little progress at finding ways to implement multiple policies in the same system. A moment's consideration will convince us that these policies are structurally different in ways that cannot be magicked away. For example, Bell-LaPadula is stateless while the others are stateful; the BMA policy pushes access control decisions to the periphery while the others centralise them; and Clark-Wilson localises security state in a different way from Chinese wall. This is inconvenient in real life. One might expect, for example, that a military hospital would want to implement something like the BMA policy for its patient records, Clark-Wilson for its accounts and Bell-LaPadula for its relationship with the outside world. But although there has been extensive discussion of the problems [11], no-one really knows how to do this.

A third problem is that many important applications do not yet have an agreed security policy model comparable to those mentioned above. The obvious example is the certification of public keys. Here we can quickly state a layman's explanation of the requirements – that *'accurate copies of the public key corresponding to a given role or entity should be made available, together with up-to-date information about keys that have been revoked'* – but so far we are not aware of anyone having formalised this in the manner of the above models.

In general, we expect that applications will continue to lead theory, and this led us to believe that rather than proposing an abstract system that could implement the existing set of models, we needed a tool that could be used for rapid prototyping of protection strategies, while hopefully also facilitating analysis and otherwise promoting good design.

Our fourth problem follows from the dual-control requirement in Clark-Wilson. We will often need to support a variety of mechanisms whereby an action is taken only if a number of other actions have been taken previously by other principals; in fact we need a general syntax to support this. Cryptologic mechanisms such as threshold signatures can cause an action to be taken if k out of n principals agree, but real life is more complicated. Threshold mechanisms can only do so much[2]. To make a real contribution to resilience and survivability in real applications, separation of duty usually needs a functional element that is bound in with the application. Thus the syntax needs to be accessible to the application developer, who must be able to make

first author's experience of banking systems, some 20–40% of the IT budget is spent on availability, 2% on audit and an insignificant amount on crypto. The second author's experience of telecommunications is that although mobile communications carriers recognise the advantages of channel encryption for communication that can easily be intercepted, the availability of the service has priority over crypto because the cost of secure communication still exceeds the loss from attacks and fraud.

[2] A useful parallel is that a disk mirroring system cannot protect users against accidentally typing `'rm *'` when in a different directory from the one they thought they were in – and, to a first approximation, all the recoveries from backup done at our site are to recover from user errors like this.

subtle decisions based on content from a number of independent sources; the cryptographic 'quick fix' of a threshold signature will usually not be enough.

Our fifth problem follows on from this point: that the computer security and cryptology communities diverged about fifteen years ago, and unfortunately many of the cryptologic mechanisms that have become standards, whether formally or otherwise, are hard to integrate with the computer security mechanisms required in real applications.

These five problems – the erroneous emphasis on confidentiality, the multipolicy problem, the need for rapid prototyping and testing of protection strategies in advance of a formal model, the need to support a variety of resilience mechanisms and the general difficulty of integrating crypto with computer security – form the backdrop for our fresh look at secure publishing.

3 The Jikzi model

The prototyping tool which we have built to investigate document security is named Jikzi, after the world's oldest publication produced using a moveable type printing press. This was printed in Korea in 1377, some 63 years before Gutenberg, and is a Buddhist religious text.

A paper which we presented at the 1999 Security Protocols workshop gives much fuller details of our Jikzi system [12]. There is also an alpha implementation, which although still under development is available online for testing and experimentation; testers are welcome [13].

Briefly, Jikzi enables a user to play with security policies by specifying style sheets which define a document type in terms not just of the data elements that must or may be present, but also of security primitives such as hashing and signature. The mechanism, Jikzi Markup Language or JML, is based on XML and allows one to define objects such as digital certificates and electronic cheques. There is also a sketch of an authentication logic, inspired by BAN, which may be used to verify a particular design constructed using these primitives.

The Jikzi model not only supports simple primitives such as the ERL model of accompanying a URL with a hash of the object one expects to find there; it also enables users to implement document types that inherit properties from other types in the manner of cascading style sheets. Thus, for example, both a cheque and a bill of lading are special cases of a bill of exchange, and a bank cheque is a special case of a cheque; so having agreed a definition of a bill of exchange, we can refine it to a cheque and then to a bank cheque. The language can be extended to support other security services such as timestamping, and registering a document to ensure its uniqueness. Examples are given in [12].

To give the reader some flavour of what JML looks like, here is an example of an electronic cheque:

```
<!-- corpCheque.xml -->
<?xml version="1.0"?>
<!DOCTYPE eCheque SYSTEM "eCheque.dtd">

<eCheque>
<dtd>
   <dtdInfo name="stdDef.dtd" version="1.0"/>
   <dtdInfo name="signList.dtd" version="1.0"/>
   <dtdInfo name="eCheque.dtd" version="1.0"/>
</dtd>
<chequeBody>
    <chequeId>00883627</chequeId>
    <account>23-45-67 1234567</account>
    <payer>University of Cambridge</payer>
    <payee foreName="William" surName="Hopkinson"/>
    <payment amount="19.95" currency="UKP"/>
    <issueDate year="1999" month="01" day="15"/>
    <notLater year="1999" month="06" day="30"/>
    <timestamp>872043082393</timestamp>
    <signInfo>
        <signer foreName="John" surName="Smith"/>
        <signAlgo algoName="PGP-RSA" version="5.5"/>
    </signInfo>
    <signInfo>
        <signer foreName="Edward" surName="Thompson"/>
        <signAlgo algoName="PGP-DSS" version="5.5"/>
    </signInfo>
</chequeBody>
<signList>
<sign>
<signInfo>
    <signer foreName="John" surName="Smith"/>
    <signAlgo algoName="PGP-RSA" version="5.5"/>
</signInfo>
<pKeyInfo>
    <pKeyVersion>1.0</pKeyVersion>
    <cert certIssuer="Cambridge Certificate Agency"
    certSerial="1234567890"
    certUrl="http://www.cca.com/certs/12345"
    revokeUrl="http://www.cca.com/revoke?sn=12345">
    <pKey>
    lQMFEDWhboWuyrPDhRvRXQEBkp4D/ivwpsci5MJQXUA
    bcPOUQquOgzMpp7W5KXP1Cit9EyqaPtet+1nkaoRXYv
    FQIB/eBjkcvNaA02w/mvHQRQYiAzz6kdPSn/rt9THkX
    LAOsOekv
    =1zy8
    <pKey>
</pKeyInfo>
<signature>
CZ/SDEjG6wt7V3uXWbZGVOpVg5LJg8j7bONjtdDuAHy
asD8dsMrWe82J23Kwe7sd2jh2348fsKS92R82kw/Tus
```

```
IyeYFI87qHE=
=OTeM
</signature>
</sign>
<sign>
<signInfo>
    <signer foreName="Edward" surName="Thompson"/>
    <signAlgo algoName="PGP-DSS" version="5.5"/>
</signInfo>
<pKeyInfo>
    <pKeyVersion>1.0</pKeyVersion>
    <cert certIssuer="Cambridge Certificate Agency"
    certSerial="2345678901"
    certUrl="http://www.cca.com/certs/23456"
    revokeUrl="http://www.cca.com/revoke?sn=23456">
    <pKey>
    SHllb24gTGVlICgxMDI0KSA8Sm9uZy1IeWVvbi5MZWV
    trSrLDLzXysRlsCHis29Q74wmeTysqY3j2z+RtzAgXb
    ErsHSe7p3Jk23Ks23RksE89wEn32Zy7gwl29rt3l9/S
    L12s7ejk
    IEz1y
    </pKey>
</pKeyInfo>
<signature>
E0a57bT2+xWWds0Jh3wpIqV25B6+ExJA6xnAB3Az5hd
XZC36FgshDjRks72EosTNmsd7Us4ePgsQ/ZeX82HHiQ
xAEALBQYiHd
n/rt9
</signature>
</sign>
</signList>
</eCheque>
<!-- end of corpCheque.xml -->
```

For document type definitions and details of other objects such as certificates, see [12].

In the rest of this paper, we will concentrate on the critical abstractions, and the lessons learned. The two critical abstractions in making the Jikzi model tractable are that persistent file stores exist, and that confidentiality concerns are limited to labeling.

3.1 Append-only file stores

In practice, many business and professional data storage systems can be modelled as append-only file stores. This holds over a large range of organisations, from banks which are required by law to retain six years' transaction history, down to a physician in private practice who uses a CD-ROM as a backup mechanism for the medical records on his PC. As for the academic literature, one of us proposed the Eternity Service – a design for a file store distributed across the net in an anonymous and almost holographic way, such that the persistence of the data could be assured even in the face of exceptionally determined attacks [14].

Append-only file stores all suffer from the theoretical vulnerability that an attacker with unrestricted write access could fill them up and thus deny the service they are designed to provide. We consider that this is not a problem in real life; a bank's mainframe operators use applications which generate a predictable quantity of customer account data, journals, audit records and so on per day, while a medical practitioner can see only so many patients and type only so many kilobytes of notes. Services made available to the public, such as backup services or conceivably Eternity servers, are charged for, and if the demand rises then more storage can be bought. So we will disregard flooding attacks.

The simplification bought by the assumption of append-only file stores means, for example, that one can sign a document simply by including it (or a hash of it) in a file designated for the purpose and to which no-one else has write access. At the theoretical level, it enables us to separate out the mechanisms designed to provide availability from those which support integrity.

3.2 Restricting consideration of confidentiality

Another great simplification is achieved by deciding to limit our consideration of confidentiality to one of maintaining the integrity of labels. Thus a designer may write a style sheet to ensure that any document initially labelled 'Top Secret' remains so, and that any newly created document incorporating it is labelled appropriately. The mechanisms whereby certain users are prohibited from viewing certain documents are considered to be a separate problem that must be solved at a higher level in the system.

At the theoretical level, assuming that confidentiality concerns are limited to the integrity of labelling means that we can separate out the mechanisms designed to provide confidentiality from those which support integrity.

3.3 A publishing security policy

The two assumptions together give us a world in which integrity is essentially our only concern. It may be compared with the world of Bell-LaPadula which focusses on confidentiality, and the discussion of availability in [15].

As an example of the value of this, we present in [12] a simple security policy model for publishing: a file store in which each user has append access to exactly one file, and all files are world readable. We show there how some simple applications fit this model, and indicate its more general usefulness. In

particular, it appears to have the power of Clark-Wilson but much greater clarity and simplicity.

We will now turn to the more general lessons that we have learned from experimenting with Jikzi.

4 Lessons learned

The lessons learned so far are largely pragmatic rather than theoretical but given the speed with which the commercial world is adopting XML and internet technology, and rebuilding business applications on top of them, they are nonetheless important enough to be worth reporting while this work is still in progress.

The first thing we have learned is that document security is much more complicated and involves much more work than simply signing a digital object with a properly certified key and leaving the reader to draw his own conclusion.

A nice example comes from the electronic version of the drug formulary mentioned above. This book, which is published every six months, contains not just a list of the drugs which a UK medical practitioner is legally allowed to prescribe, but also information such as dosage per bodyweight and cross reactions – data that must be interpreted accurately by applications in future hospital systems if the full safety and other benefits are to be extracted.

Furthermore, the book is viewed through a number of customisable filters. The typical hospital has its own drug formulary, which reflects local policy on issues such as when generic drugs must be prescribed instead of more expensive branded products, or where drugs seen as dangerous or difficult are restricted to named doctors. In the current implementation, the national formulary text is displayed in black, standard headlines in blue, and local hospital restrictions in red [2]. This is simple enough; but future editions are likely to have further filters (e.g. for local groups of family doctors who do follow-up care for hospital patients) and there may also be experimental drugs that are not yet in the national formulary but are available under restricted conditions.

There are many parallels in commercial applications. For example, the bundle of documents that enables a company to collect payment for a shipment of goods may include a letter of credit, an invoice, an insurance certificate and an inspection certificate – all of which come from different sources and are seen through different filters by different parties to the transaction. Here, too, the potential complexity growth appears to be without limit, especially where there are intermediate processing steps such as message processing systems.

Human computer interfaces may start to pose further problems. In other applications, content produced in colour rather than monochrome may preclude the use of text colour as a means of indicating the source of information. Part of the answer may be colouring the browser frame, an approach taken by Trusted X [16]. How well this can be squared with the paragraph-level labelling favoured by some government agencies, and implicit in the complex structure of documents such as drug formularies, remains to be seen.

The second lesson learned is that designing security in XML is not necessarily easier than in any other environment. Application security often has an inherent complexity, which can be shifted to one place or another but still has to be tackled in the end. We note the opinion expressed by the perennial IT optimists that the huge costs of making business systems communicate with each other will somehow magically vanish when the acronym 'EDI' is replaced by the acronyms 'XML' and 'extranet'. We beg to differ. Getting a variety of different systems to recognise that a given field represents a drug dosage, and that the unit is milligrams rather than micrograms, has inherent complexity and criticality.

The third lesson is that tools can still help. Inheritance is important, and style sheets can help (so long, of course, that we do not end up with a proliferation of incompatible ones). It will also be important to be able to draw on existing work; it would be pointless, for example, to discard all the work that has been done on healthcare messaging standards such as HL7 and reinvent the wheel all over again. So although much standards work may need to be re-done, one might hope that it will be done more quickly and thoroughly the next time around (even though a pessimist will fear that backward compatibility will turn out to be a millstone). The standards task will involve the markup languages themselves; it would be ideal if we could merge the existing security markup systems into a single language that is powerful enough for all requirements, and we have tried to move in this direction with Jikzi and JML. Even if the eventual market outcome falls short of this ideal, we hope that our insights may be useful.

Finally, we have learned a lot from considering a real world problem, namely the publication of medical data, and much of what we learned was not at all obvious when we set out. The experience of the Wax–ERL–Jikzi thread of research supports the view that in order to understand what new kinds of

systems are likely to require in the way of services, it is important to build prototypes and to exercise them on real applications. As often in systems research, the science continues to lag the engineering. Empirical input continues to be essential, and we encourage readers to apply our ideas and tools to their own areas of interest.

5 Conclusions

The three main aspects of electronic commerce that need to be protected against error and attack are publication, payment and copy control. Payment and copy control are each the subject of hundreds of research papers, while publication has been largely ignored.

In this paper we discussed the issues as we have come to understand them from undertaking a series of projects that were largely inspired by practical problems of medical publishing. We hope that we have managed to convince the reader that the authentication of published material is a problem worthy of serious study; it is not just much more complex than it seems, but central to electronic commerce.

6 Acknowledgement

The second author was supported by EPSRC grant number GR/L95809 on Resilient Security Mechanisms from 4/98 until 10/99. Since February 2000 the work described in this paper has been continued by Filonet Korea Inc.

References

[1] RJ Anderson, V Matyáš, FAP Petitcolas, IE Buchan, R Hanka, "Secure books: protecting the distribution of knowledge", in *Security protocols: proceedings of the 5th international workshop*, Springer-Verlag LNCS v 1361 (1997) pp 1–12; `http://www.cl.cam.ac.uk/ftp/users/rja14/wax.ps.gz`; the Wax home page is at `http://www.medinfo.cam.ac.uk/wax/`

[2] British Medical Association and Royal Pharmceutical Society of Great Britain, *British National Formulary*, ISBN 0 85369 434 6; electronic version published by Pharmaceutical Press

[3] RJ Anderson, V Matyáš, FAP Petitcolas, "The Eternal Resource Locator: an alternative means of establishing trust on the world wide web", in *3rd USENIX workshop on electronic commerce* (1998), pp 141–153; `http://www.cl.cam.ac.uk/~fapp2/papers/ec98-erl/`

[4] T Bray, J Paoli, CM Sperberg-McQueen, *'Extensible Markup Language (XML 1.0)'*, W3C Recommendation REC-xml-1998210, World Wide Web Consortium, Feb 1998 `http://www.w3.org/TR/REC-xml`

[5] RJ Anderson, "The Risks and Costs of UK Escrow Policy", in *UK House of Commons Trade and Industry Committee*, seventh report, session 1998–99; HM Stationery Office, 12 May 1999, pp 164–169; `http://www.parliament.the-stationery-office.co.uk/pa/cm199899/cmselect/cmtrdind/187/18702.htm` and `http://www.cl.cam.ac.uk/users/rja14/dtiselcom.html`

[6] RJ Anderson, "Why Cryptosystems Fail", in *Communications of the ACM* v 37 no 11 pp 32–40, ACM, November 1994; `http://www.cl.cam.ac.uk/users/rja14/wcf.html`

[7] J Kravitz, "SDML – Signed Document Markup Language", W3C Note NOTE-SDML-19980619, World Wide Web Consortium, June 1998; `<http://www.w3.org/TR/1998/NOTE-SDML-19980619/>`

[8] YH Chu, P DesAutels, B LaMacchia, P Lipp, *'PICS Signed Labels (DSig) 1.0 specification'*, W3C Recommendation REC-DSig-label-19980527, World Wide Web Consortium, May 1998; `http://www.w3.org/TR/REC-DSig-label`

[9] E Amoroso, *'Fundamentals of Computer Security Technology'*, Prentice Hall, 1994

[10] RJ Anderson, "A Security Policy Model for Clinical Information Systems", in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp 30–43, Oakland, CA, 1996; conference paper is `http://www.cl.cam.ac.uk/ftp/users/rja14/oakpolicy.ps.Z`; full BMA version is `http://www.cl.cam.ac.uk/users/rja14/policy11/policy11.html`

[11] HH Hosmer, "The Multipolicy Paradigm for Trusted Systems", in *New Security Paradigms Workshop 1993*, pp 19–32, Little Compton, RI, 1993.

[12] RJ Anderson, JH Lee, "Jikzi: A New Framework for Secure Publishing", *to appear in Security Protocols: Proceedings of the 5th international workshop*, to be published by Springer-Verlag in the LNCS series; `http://www.cl.cam.ac.uk/~jhl21/jikzi-cpw/`

[13] JH Lee, Jikzi system; if you want to be a tester, please contact the second author

[14] RJ Anderson, "The Eternity Service", in *Proceedings of Pragocrypt 96*, pp 242–252, GC UCMP, ISBN 80-01-01502-5, Prague, 1996; `http://www.cl.cam.ac.uk/ftp/users/rja14/eternity.ps.Z`

[15] RM Needham, "Denial of Service", *Communications of the ACM* v 37 no 11 (Nov 94) pp 42–46

[16] J Epstein, R Pascale, "User Interface for a High Assurance Windowing System", in *Security Applications 93* pp 256–264