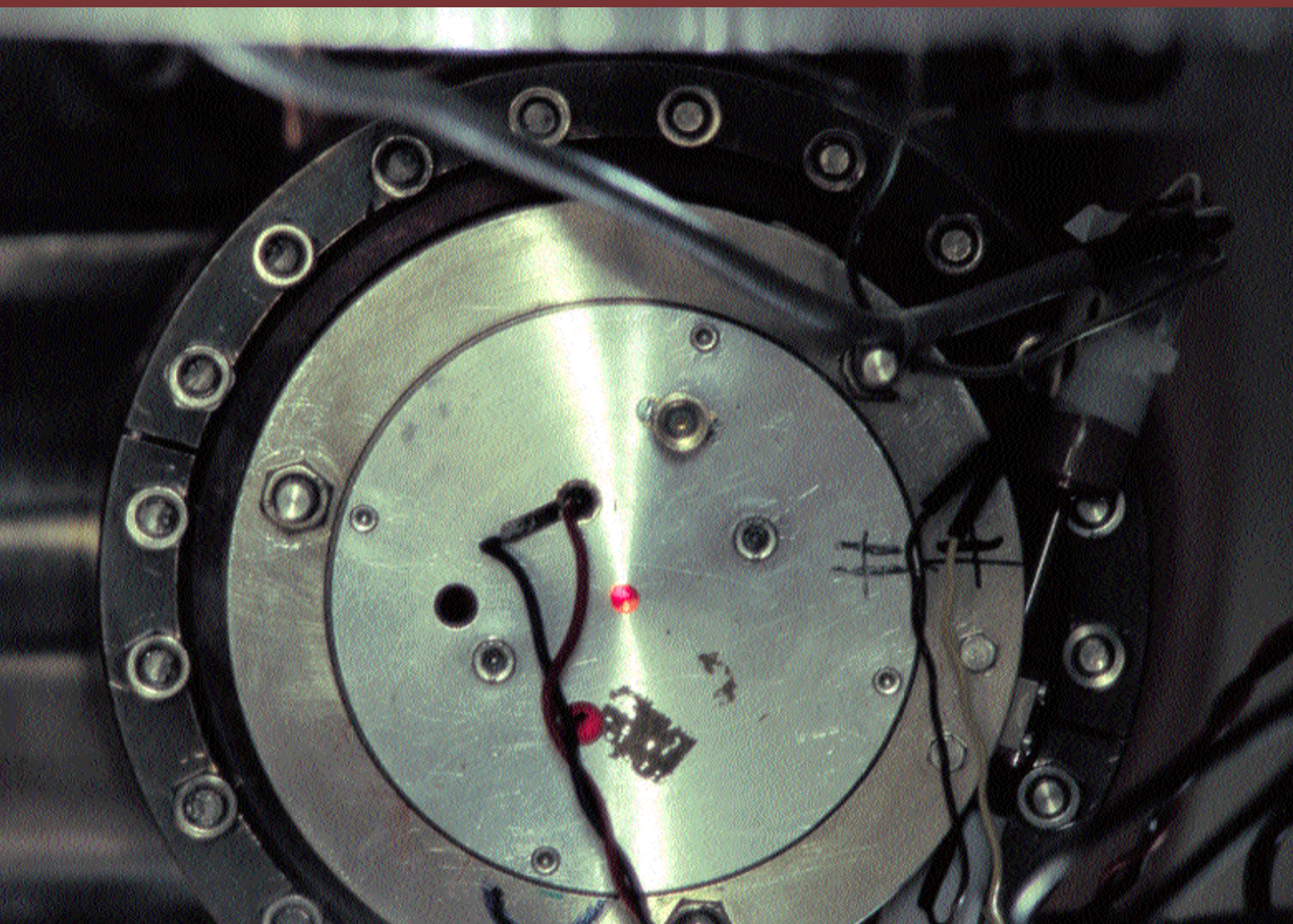
PRICEWATERHOUSECOOPERS

Issue 3 1999

CCE Quarterly Journal

Cryptographic Centre of Excellence



CERTIFICATION SERVICE

FOR ELECTRONIC SUBMISSION OF PROPOSALS UNDER THE FIFTH FRAMEWORK PROGRAMME

fp5-csp.org

A new way to submit proposals.
Develop and submit your proposals without any paper exchange!

- Easier and quicker communication
- Reliable and fast delivery of documents
- Rapidly confirmed receipt
- No more multiple hard copies
- No mailing costs
- Critical time saving
- Free of charge

Get a free digital certificate to exploit these advantages.

Certification guarantees that the proposals are:
- Confidential (through encryption)
- Legally valid (with a digital signature)

The digital certificate can be requested from the FP5-CSP Consortium using ProTool, the electronic submission tool provided by the European Commission. ProTool can be obtained from Cord is web site: www.cordis.lu

EUROPEAN COMMISSION

Contact information
For more information and support on digital certificates visit the web: **WWW.FP5-CSP.ORG**
or send an e-mail to **info@fp5-csp.org**

GlobalSign

PRICEWATERHOUSECOOPERS

BALTIMORE

Issue 3 1999

CCE Quarterly Journal

Cryptographic Centre of Excellence

The PricewaterhouseCoopers Cryptographic Centre of Excellence (CCE) was formed by the company's Global Risk Management Solutions practice in order to unite members from around the globe with unique expertise in cryptography and cryptographic services. This was done to build a network of highly skilled professionals who could assist clients and one another throughout an engagement's lifecycle. By establishing relationships with academic institutions, leading security vendors, cryptographic research organisations and leading cryptographers, we are in a truly unique position to offer our global clients the best solutions for their cryptographic security needs.

Global Risk Management Solutions, part of PricewaterhouseCoopers, has nearly 5,000 professionals world-wide, many of them industry specialists, and offers a comprehensive identification of risks, whether they are strategic, financial or operational in nature. Our solutions-based risk identification and analysis offers guidance on industry best practices and common training programmes, using state-of-the-art methodologies and tools consistently.

By addressing the changing needs of today's business leaders, we are able to help organisations identify, assess and manage complex issues and risks across the whole enterprise - or within in any part of it - whether they are strategic, financial or operational in nature. We help clients to develop risk management solutions that minimise hazard, resolve uncertainty and maximise opportunity.

Contact Information

Dr. Alastair MacWillson

Partner in PwC London and global leader for Technology Risk Services within PwC's Global Risk Management Solutions practice

alastair.macwillson@uk.pwcglobal.com

Geoffrey C. Grabow CISSP

Leader of PwC's Cryptographic Centre of Excellence

geoffrey.grabow@uk.pwcglobal.com

John Velissarios M.Comp.Sci.

Manager, Cryptographic Centre of Excellence

john.velissarios@uk.pwcglobal.com

More information and previous editions of the Quarterly Journal can be found at :

www.pwcglobal.com/cce

To SUBSCRIBE to the weekly e-mail cryptographic news, send a message to either Geoffrey or John at the addresses given above.

The views expressed in this publication are not necessarily the views of PricewaterhouseCoopers.

In this Issue

Selecting Cryptographic Key Sizes in Commercial Applications

by Arjen K. Lenstra, Citibank & Eric R. Verheul, PricewaterhouseCoopers

Cryptography is an important tool in the protection of e-commerce applications, and, more specifically, is used to protect the confidentiality, integrity, authenticity and non-repudiation of information. In the end, the protective quality depends not only on the cryptographic technology and the key sizes that are applied but also, and in particular, on the way in which this technology is implemented (protocol design).

In this article we present guidelines for the determination of cryptographic key sizes; other major issues such as protocol design will not be discussed. This article is a summary of [2], in which we present a more detailed substantiation of our guidelines. Recommendations on key sizes can be found in a variety of sources, such as cryptographic literature or vendor documentation. Unfortunately it is often hard to tell on what premises those recommendations are based. As far as we know this article is the first uniform, clearly defined, and properly documented treatment of this subject. Our guidelines will enable organisations to arrive at a balanced evaluation of key size aspects in the purchase or development of cryptographic applications. They have been formulated with reference to the main cryptographic primitives, being:

- Symmetric key systems, eg the Data Encryption Standard (DES);
- Classical asymmetric (or public) key systems, being the RSA system and the traditional discrete logarithm systems, such as ElGamal (Elg) and Diffie-Hellman (DH). All of these are supported in the popular encryptor known as 'Pretty Good Privacy' (PGP);
- Subgroup discrete logarithm systems, including the US Digital Signature Algorithm (DSA) and the Schnorr digital signature system; and
- Elliptic Curve systems.

In addition to featuring in brochures, these systems are mentioned in the set of export control regulations known as the Wassenaar Arrangement and issued in order to reduce the proliferation of (powerful) cryptographic products. We will briefly discuss these systems in the appendix, including reference to the maximum key sizes for cryptographic products that do not require an export licence.

We are slightly hesitant about providing these key size guidelines. Organisations looking for a reliable system tend to be more focused on the cryptography and key sizes used than on the design in which the technology is deployed. Experience has taught us, however, that failures in cryptography almost invariably originate in some design error within the system as a whole,

rather than in a wrong choice of cryptosystem or key size (also see [1]). In other words, it is better to concentrate on the quality of the overall design than to be fixated on the technology or key sizes used. Two examples may illustrate our point. The cryptography and key sizes used by the PGP encryptor mentioned above offer a perfectly acceptable level of security for information transmitted via the Internet. But the user-password that protects the private PGP keys stored on an Internet-accessible PC does not necessarily offer the same security. Even if the user is highly security-conscious and selects a random password consisting of 8 characters from a set of 128 choices, the resulting level of security is comparable to the protection offered by the recently broken 'Data Encryption Standard' (see [3]), and thereby unacceptable by today's standards.

An even more disturbing example can be found in many network configurations. There, each user may select a password that consists of 14 characters, which should, in principle, offer enough security. Before transmission over the network the passwords are encrypted, with the interesting feature, however, that each password is split into two parts of at most 7 characters each, and that each of the two resulting parts is treated separately, ie encrypted and transmitted over the network. This effectively reduces the password length of 14 to 7, offering a level of security that clearly falls short of current standards.

Experience has shown that a single bright idea may prove that a particular cryptographic protocol is considerably less secure than expected.

Our suggestions are based on reasonable extrapolations of developments that have taken place during the last two decades. This approach may fail: experience has shown that a single bright idea may prove that a particular cryptographic protocol is considerably less secure than expected.

This article is structured as follows:

- In Section 2 we discuss our model for the selection of key sizes;
- In Section 3 we discuss the results generated by the model and their consequences; and
- In Section 4 we give further comments on our model.

The model

As soon as any reasonable doubts about the quality of the system's design have been dispelled, ie as soon as it is clear that the system can only be violated by means of a direct attack on the cryptography used, the choice of key size must be made. This choice primarily depends on the following three quantifiable parameters:

I. *Life span:* the expected time the information needs to be protected against attacks;

II. *Security margin:* an acceptable degree of certainty that any attacks will prove unfeasible during the life span of the information. This largely depends on the *identity* of the attacker and the computational and financial power of their attack; and

III. *Cryptanalysis:* the effectiveness of attacks during the life span of the information.

1. Life span

This is the crucial parameter within our model. It is the user's responsibility to evaluate until what year the protection should be effective.

2. Security margin

In practice it proves to be very difficult to identify the attackers of an organisation and its information. It is even harder to gauge the power of the attacker once its identity has been established. This means that it is virtually impossible to quantify security margins in this way. We have therefore mapped out a different approach in which we select a security margin from the past and extrapolate it to the future using two other hypotheses.

Hypothesis I
The basic assumption underlying our extrapolations is that the Data Encryption Standard (DES) was sufficiently secure for commercial applications until 1982, given that it was introduced in 1977 and stipulated to be reviewed every five years. We therefore assume that the computational effort required for breaking DES offered an adequate security margin for commercial applications up to 1982.

The computational effort required to break DES is estimated to amount to $5*10^5$ Mips Years (see [2]). One Mips Year is the amount of computation that can be performed in one year by a single VAX 780, and is roughly equivalent to 20 hours on a 450MHz PentiumII processor. Thus, $5*10^5$ Mips Years is roughly 14,000 months on a 450MHz PentiumII processor, or 2 months on 7000 of such processors.

Given that computers have become both faster and cheaper over the years, this computational effort must be extrapolated to the present and the future. For this purpose we use a second hypothesis: Moore's Law.

Hypothesis II
According to an internationally accepted interpretation of Moore's Law, the computational power of one chip doubles every 18 months as new types are released. There is some scepticism as to whether this law is tenable, because fundamentally new technologies will eventually have to be developed to keep up with it. This is one of the reasons we hypothesise a variation of Moore's Law that is less technology dependent and has so far proved to be sufficiently accurate: every 18 months the amount of computing power available for one dollar doubles. It follows that the same investment will generate a factor of 210*12/18 » 100 more computing power every 10 years.

Hypothesis III
Our version of Moore's Law implies that we have to consider how budgets may change over time. The US Gross National Product shows a trend of doubling every ten years: $1,630 billion in 1975, $4,180 billion in 1985, and $7,269 billion in 1995. This leads to the hypothesis that the budgets of organisations - including the ones breaking cryptographic keys - double every ten years.

Illustration: combination of Hypotheses I, II & III
If $5*10^5$ Mips Years provided an adequate security margin for commercial applications in 1982, $1*10^8$ ($\approx 2*100*5*10^5$) Mips Years will do so in 1992, $2*10^{10}$ ($\approx 200*1*10^8$) Mips Years in 2002 and $4*10^{12}$ Mips Years in 2012.

3. Cryptanalysis

Hypothesis IV
For each of the four cryptographic systems central to this article, attacks are described in the cryptographic literature. By measuring the complexity of those attacks we can establish the connection between key size and computational effort and, hence, the security margin, for each of these four cryptographic systems (see [2] for details).

It is impossible to say exactly how cryptanalysis will develop in the future. It is reasonable to assume, however, that the pace of future cryptanalytic progress is not going to vary dramatically compared with what we have seen from 1970 until 1999. For classical asymmetric systems the effect of cryptanalytic developments is similar to Moore's Law, ie, we may expect that 18 months from now an attack on such a system will require only half the computational power that would be required today. For all other systems we assume that no substantial cryptanalytic developments will take place, with the exception of systems based on elliptic curves, for which we use two types of extrapolations: no progress and progress à la Moore.

Results of the model

Our model makes it relatively easy to make predictions about key sizes based on life span, since the hypotheses, in combination with life span data, enable us to determine the security margin in Mips Years that the cryptographic system is to provide. Moreover, hypothesis IV and the life span data enable us to determine, for every identified cryptosystem, the key size that corresponds to the security margin. The key sizes are listed in table 1.

Practical consequences of the model

Use of the table

Assuming the reader agrees with our hypotheses, table 1 can be used as follows in the selection of key size. Suppose a commercial application is developed within which the confidentiality or integrity of the electronic information has to be guaranteed for 20 years, ie until 2020. The corresponding row for 2020 in table 1 shows that $2.94*10^{14}$ Mips Years can be regarded as a sufficient security margin for that information, and that the following key sizes should be considered:

- Symmetric keys of *at least* 86 bits;
- RSA moduli of *at least* 1881 bits;
- Subgroup discrete logarithm systems with group primes of *at least* 151 bits and basic primes of *at least* 1881 bits; and
- Elliptic Curve systems of *at least* 161 bits if no cryptanalytic progress is expected in this field, and *at least* 188 bits to obviate any eventualities.

Consequences for the US digital signature standard/algorithm

The American standard for digital signatures (DSS/DSA) is based on a Subgroup Discrete Logarithm system in which 160-bit subgroups are used in combination with a prime number p between 512 and 1024 bits. From our table it

Table 1

Suggested lower bounds for key sizes in bits, assuming cryptanalytic progress à la Moore affecting classical asymmetric systems

Year	Symmetric Key Size (bits)	Classical Asymmetric Key Size (RSA, Elg, DH) (bits)	Subgroup Discrete Logarithm Key Size (DSA, Schnorr)(bits)	Elliptic Curve Key Sizes (in bits)		Security Margin (Mips Years)	Corresponding no. of Years on 450MHz PentiumII PCs	Corresponding (minimal) Budget for Attack in 1 Day (USD)
				Progress				
				no	yes			
1982	56	417	102	105		$5.00 * 10^5$	$1.11 * 10^3$	$3.98 * 10^7$
1985	59	488	106	110		$2.46 * 10^6$	$5.47 * 10^3$	$4.90 * 10^7$
1990	63	622	112	117		$3.51 * 10^7$	$7.80 * 10^4$	$6.93 * 10^7$
1995	66	777	118	124		$5.00 * 10^8$	$1.11 * 10^6$	$9.81 * 10^7$
2000	70	952	125	132	132	$7.13 * 10^9$	$1.58 * 10^7$	$1.39 * 10^8$
2001	71	990	126	133	135	$1.21 * 10^{10}$	$2.70 * 10^7$	$1.49 * 10^8$
2002	72	1028	127	135	139	$2.06 * 10^{10}$	$4.59 * 10^7$	$1.59 * 10^8$
2003	73	1068	129	136	140	$3.51 * 10^{10}$	$7.80 * 10^7$	$1.71 * 10^8$
2004	73	1108	130	138	143	$5.98 * 10^{10}$	$1.33 * 10^8$	$1.83 * 10^8$
2005	74	1149	131	139	147	$1.02 * 10^{11}$	$2.26 * 10^8$	$1.96 * 10^8$
2006	75	1191	133	141	148	$1.73 * 10^{11}$	$3.84 * 10^8$	$2.10 * 10^8$
2007	76	1235	134	142	152	$2.94 * 10^{11}$	$6.54 * 10^8$	$2.25 * 10^8$
2008	76	1279	135	144	155	$5.01 * 10^{11}$	$1.11 * 10^9$	$2.41 * 10^8$
2009	77	1323	137	145	157	$8.52 * 10^{11}$	$1.89 * 10^9$	$2.59 * 10^8$
2010	78	1369	138	146	160	$1.45 * 10^{12}$	$3.22 * 10^9$	$2.77 * 10^8$
2011	79	1416	139	148	163	$2.47 * 10^{12}$	$5.48 * 10^9$	$2.97 * 10^8$
2012	80	1464	141	149	165	$4.19 * 10^{12}$	$9.32 * 10^9$	$3.19 * 10^8$
2013	80	1513	142	151	168	$7.14 * 10^{12}$	$1.59 * 10^{10}$	$3.41 * 10^8$
2014	81	1562	143	152	172	$1.21 * 10^{13}$	$2.70 * 10^{10}$	$3.66 * 10^8$
2015	82	1613	145	154	173	$2.07 * 10^{13}$	$4.59 * 10^{10}$	$3.92 * 10^8$
2016	83	1664	146	155	177	$3.51 * 10^{13}$	$7.81 * 10^{10}$	$4.20 * 10^8$
2017	83	1717	147	157	180	$5.98 * 10^{13}$	$1.33 * 10^{11}$	$4.51 * 10^8$
2018	84	1771	149	158	181	$1.02 * 10^{14}$	$2.26 * 10^{11}$	$4.83 * 10^8$
2019	85	1825	150	160	185	$1.73 * 10^{14}$	$3.85 * 10^{11}$	$5.18 * 10^8$
2020	86	1881	151	161	188	$2.94 * 10^{14}$	$6.54 * 10^{11}$	$5.55 * 10^8$
2021	86	1937	153	163	190	$5.01 * 10^{14}$	$1.11 * 10^{12}$	$5.94 * 10^8$
2022	87	1995	154	164	193	$8.52 * 10^{14}$	$1.89 * 10^{12}$	$6.37 * 10^8$
2023	88	2054	156	166	197	$1.45 * 10^{15}$	$3.22 * 10^{12}$	$6.83 * 10^8$
2024	89	2113	157	167	198	$2.47 * 10^{15}$	$5.48 * 10^{12}$	$7.32 * 10^8$
2025	89	2174	158	169	202	$4.20 * 10^{15}$	$9.33 * 10^{12}$	$7.84 * 10^8$
2026	90	2236	160	170	205	$7.14 * 10^{15}$	$1.59 * 10^{13}$	$8.41 * 10^8$
2027	91	2299	161	172	207	$1.21 * 10^{16}$	$2.70 * 10^{13}$	$9.01 * 10^8$
2028	92	2362	162	173	210	$2.07 * 10^{16}$	$4.59 * 10^{13}$	$9.66 * 10^8$
2029	93	2427	164	175	213	$3.52 * 10^{16}$	$7.81 * 10^{13}$	$1.04 * 10^9$
2030	93	2493	165	176	215	$5.98 * 10^{16}$	$1.33 * 10^{14}$	$1.11 * 10^9$
2031	94	2560	167	178	218	$1.02 * 10^{17}$	$2.26 * 10^{14}$	$1.19 * 10^9$
2032	95	2629	168	179	222	$1.73 * 10^{17}$	$3.85 * 10^{14}$	$1.27 * 10^9$
2033	96	2698	169	181	223	$2.95 * 10^{17}$	$6.55 * 10^{14}$	$1.37 * 10^9$
2034	96	2768	171	182	227	$5.01 * 10^{17}$	$1.11 * 10^{15}$	$1.46 * 10^9$
2035	97	2840	172	184	230	$8.53 * 10^{17}$	$1.90 * 10^{15}$	$1.57 * 10^9$
2036	98	2912	173	185	232	$1.45 * 10^{18}$	$3.22 * 10^{15}$	$1.68 * 10^9$
2037	99	2986	175	186	235	$2.47 * 10^{18}$	$5.49 * 10^{15}$	$1.80 * 10^9$
2038	99	3061	176	188	239	$4.20 * 10^{18}$	$9.33 * 10^{15}$	$1.93 * 10^9$
2039	100	3137	178	189	240	$7.14 * 10^{18}$	$1.59 * 10^{16}$	$2.07 * 10^9$
2040	101	3214	179	191	244	$1.22 * 10^{19}$	$2.70 * 10^{16}$	$2.22 * 10^9$

follows that the security offered by DSS/DSA becomes doubtful after 2002, which is unacceptable as it is essential for digital signatures to have a considerable life span. The table shows that if their reliability is to be ensured until 2026, it is wiser to use DSA with 2236-bit prime numbers (considerably above the DSA maximum of 1024 bits). Note that this does not add to the length of the signature.

Consequences for international SSL versions

The Secure Sockets Layer (SSL) protocol is a popular protocol for the exchange of confidential information (credit card numbers and the like) between a web browser (= customer) and webserver (= e-commerce shopkeeper). SSL uses an RSA key placed on the webserver (Microsoft Internet Information Server, Netscape Enterprise Server, Apache Server). The key is usually a certificate, ie signed by a Certificate Authority. The RSA key enables the exchange of a session key between the browser and the webserver which is used to encrypt confidential information. This means that the connection between browser and server is secure only if both the session key and the RSA modulus are sufficiently large.

Due to the Wassenaar Arrangement, web*browser* versions that are internationally available use key sizes of only 40 bits. This is insufficient with respect to current standards

(so small, in fact, as to have been left out of table 1). In web*server* versions that are internationally available (frequently used in Europe) RSA moduli of only 512 bits are used. This, too, falls short of today's standards. This is because any attacker that manages to break this SSL RSA key will be able to access all session keys, and hence all the information encrypted by those keys. Our table shows that the level of security provided by 512-bit RSA moduli had already become insufficient in 1990, but in spite of that international versions of webservers, and hence the 512-bit RSA moduli, continue to be widely used. In 1999, scientists made the first move towards factorisation of a 512-bit modulus. They reached their goal on 22 August of that year. This means that in addition to direct security risks, publicity risks are involved in the use of 512-bit RSA moduli, since the organisations that use them may receive a bad press now that 512-bit RSA moduli have been reported to be unsafe.

The limit in the Wassenaar Arrangement for symmetrical encryption is 64 bits, which offers more protection than the 56 bits of DES. The table above shows that at the present moment the level of security offered by 64-bit symmetrical encryption is roughly equivalent to the protection offered by 768-bit RSA. It would be logical, therefore, for the limit for RSA keys in the Wassenaar Arrangement to be set at 768 bits. This could considerably raise the level of security offered by international implementations of SSL.

American ("Domestic") webservers that use safer key sizes (eg 1024 bits) require an American export licence. Until very recently only banks qualified for such a licence, but in principle insurance companies, medical institutions and on-line merchants now qualify as well for a domestic webserver export licence.

Critical comment: software versus hardware attacks

We have presented key size recommendations for several different cryptographic systems. For a certain specified level of security these recommendations may be expected to be equivalent in the sense that the computational effort or number of Mips Years for a successful attack is more or less the same for all cryptographic systems under consideration. So, from a computational point of view the different cryptographic systems offer more or less equivalent security when the recommended key sizes are used. This *computationally equivalent security* should not be confused with, and is not necessarily the same as, *equipment cost equivalent* security, or *cost equivalent* security for short. Here we say that two systems offer cost equivalent security if accessing or acquiring the hardware that allows a successful attack in a certain fixed amount of time costs the same amount of dollars for both systems. Note that although the price is the same, the hardware required may be quite different for the two different attacks; some attacks may use multi-purpose (eg PCs), for other attacks it may be possible to get the required Mips Years relatively cheaply by using special-purpose hardware. Following our guidelines does *not* necessarily result in cost equivalent security. The most important reason why we have opted for computationally equivalent security as opposed to cost equivalent security, is that we found that computational equivalence allows rigorous analysis, mostly independent of our own judgement or preferences. Analysis of cost equivalence, on the other hand, depends on choices that are rather subjective, can change over time, and have a considerable effect on the outcome.

The level of security provided by 512-bit RSA moduli had already become insufficient.

It is indicated though in [2] that, apart from the classical asymmetric key, for all cryptographic systems central to this article, the cost per Mips Year for special-purpose breaking hardware roughly coincides. The required budget for generating the security margin (in Mips Years) of a given year for these systems is given in the last column of table 1. Moreover, it is indicated in [2] that special-purpose breaking hardware for the classical asymmetric key systems currently seems to be more expensive; a factor 2500 is a rough estimation. This means that if one is interested in cost equivalence instead of computational equivalence, using this factor and taking the cost of breaking hardware different from the classical asymmetric systems as a basis, then for the year y one has to consider the classical asymmetric key sizes of the year y-8. Moreover, the subgroup discrete logarithm key size that is based on *this* asymmetric key size should be taken 2 bits longer than indicated in the year y. This is to compensate for the fact that multiplications based on this smaller asymmetric key size, require less computational effort. For the above-mentioned reasons we advise against indiscriminate use of the resulting smaller key sizes.

About the authors

Arjen K. Lenstra works at the Emerging Technologies Department of Citibank's Corporate Technology Office in New York. Dr Lenstra has acquired an international reputation as an expert in the field of cryptanalysis. For example, the well-known RSA-129 challenge was broken using his software.
E-mail: Arjen.Lenstra@citicorp.com

Eric R. Verheul works for PriceWaterhouseCoopers in Utrecht (the Netherlands). He offers consultancy services on information security for new e-commerce applications in particular, and is scientifically involved in both theoretical and applied cryptology. Dr Verheul is also a lecturer in information security at Eindhoven University of Technology.
E-mail: Eric.Verheul@nl.pwcglobal.com

Neither the authors nor their employers accept any liability for the use of the key sizes as recommended in this article. The contents of this article are the sole responsibility of its authors and not of their employers. The authors do not have any financial or other material interests in the conclusions attained in this paper, nor were they inspired or sponsored by any party with commercial interests in cryptographic key size selection.

The data presented in this article were obtained in a two stage approach that was strictly adhered to: formulation of the model and collection of the data points, followed by computation of the lower bounds. No attempt has been made to alter the resulting data so as to better match the authors' (or any other person's) expectations or preference. The authors have made every attempt not to be biased towards their personal favourite cryptosystems, if any. Although the analysis and the resulting guidelines seem to be quite robust, this will no longer be the case if there is some 'off-the-chart' cryptanalytic or computational progress affecting any of the cryptosystems considered here. Indeed, according to one of the present authors, strong long-term reliance on any current cryptosystem without very strong physical protection of all keys involved - including public ones - is irresponsible.

References

[1] Why Cryptosystems fail, R.J. Anderson, Communications of the ACM, v. 37, no.11, Nov. 1994, pp. 32-40.

[2] Selecting Cryptographic Key Sizes, A.K. Lenstra, E.R. Verheul, accepted for presentation at the 2000 International Workshop on Practice and Theory in Public Key Cryptography (PKC2000), Melbourne, Australia, January 2000

[3] Cracking DES, Electronic Frontier Foundation, O'Reilly, July 1998.

Appendix

A summary description of the cryptographic primitives

The Co-ordinating Committee for Multilateral Export Controls (COCOM) was an international organisation regulating the export of strategic products, including cryptographic products, from member countries to countries jeopardising their national security. Member countries, eg European countries and the US, implemented the COCOM regulations in national legislation.

The Wassenaar Arrangement is a follow-up of the COCOM regulations and includes fairly detailed restrictions with respect to cryptography. For four types of cryptographic primitives the maximum key sizes are mentioned in respect of which no export license is required. In this article we limit ourselves to these four cryptographic primitives. Due to the nature of the Wassenaar Arrangement, it is hardly surprising that that these key sizes do not provide adequate protection in the majority of commercial applications.

It is hardly surprising that these key sizes do not provide adequate protection.

Two general types of cryptographic primitives can be distinguished: symmetric (or secret) and asymmetric (or public) key cryptosystems. Such systems are instrumental in building e-commerce enabling solutions and can be used to achieve confidentiality, integrity, authenticity, and non-repudiation of electronic information. For the sake of simplicity we will assume that there are two communicating parties, a sender S and a receiver R, who want to secure the confidentiality of their communication.

Symmetric key systems

Description: In symmetric key cryptosystems S and R share a key. To maintain confidentiality the key should be kept secret. The crucial parameter in symmetric cryptosystems is the size of the key, ie its number of bits, which depends on the type of symmetric key system used.

The best-known symmetric system is the Data Encryption Standard (DES), introduced in 1977, with a key size of 56 bits. Other examples include:

- Three Key Triple DES (key size 168, effective key size 112);
- IDEA (key size 128);
- RC5 (variable key size); and
- The future successor of DES, the Advanced Encryption Standard (AES), with key sizes of 128, 192 or 256 bits.

Wassenaar Arrangement: The maximum symmetric key sizes allowed by the Wassenaar Arrangement are 56 and 64 bits for niche market and mass market applications, respectively. The reason for this difference in key size is obvious.

Asymmetric key systems

In asymmetric key cryptosystems the receiver R has a private key (which R keeps secret) and a corresponding public key that anyone, including S, has access to. The sender S uses R's public key to encrypt information intended for R, and R uses its private key to decrypt the encrypted message. If the private key can be derived from the public key, then the system can be broken.

The nature of the private and public keys and the effort required to break the system depend on the type of asymmetric key cryptosystem. For cryptanalytic and historic reasons we distinguish the following three types:

- Classical asymmetric systems;
- Subgroup discrete logarithm systems; and
- Elliptic Curve systems.

Classical asymmetric systems

These refer to RSA and traditional discrete logarithm (TDL) systems.

In RSA the public key contains a large number, the so-called RSA modulus, which is the product of two large prime numbers. The details of the asymmetric encryption technique are beyond the scope of this article. If these two primes can be retrieved from their product, the private key can be found, thereby breaking the system. Thus, the security of RSA is based on the difficulty of the integer factorisation problem. The size of an RSA key refers to the bit-length of the modulus.

The difficulty of the so-called discrete logarithm problem in specific 'groups' serving as a basis of cryptosystems is comparable to the factorisation problem, although it falls beyond the scope of this article. The security of such systems hinges upon:

- The structure of the group; and
- The size of the group, i.e. the number of elements in it.

In a TDL system the structure of the group and the cryptosystem are based on "modulo calculating a basic prime p". The size of the group is equal to p-1. The size of a TDL key refers to the bit-length of the basic prime p. Examples of TDL systems are ElGamal (Elg) and Diffie-Hellman (DH) systems, both supported in Pretty Good Privacy.

Wassenaar Arrangement: Within the Wassenaar Arrangement the maximum key size for RSA and TDL systems is fixed at 512 bits, which means that the RSA modulus mentioned above and the basic prime must be smaller than 2512. A popular standard for both sizes is 1024 bits.

Subgroup discrete logarithm systems

Subgroup discrete logarithm (SDL) systems closely resemble traditional discrete logarithm systems, using the same structure for the group construction based on the basic prime p. However, SDL systems only use part of the group, a subgroup. The size of the subgroup is prime shared by p-1 and indicated by q. Attacks mounted against TDL systems are also effective against SDL systems. However, some attacks on SDL systems are particularly effective if the group prime q is relatively small. The key size of an SDL system refers to the bit-length of the basic prime p and the group prime q.

Wassenaar Arrangement: The Wassenaar Arrangement does not prescribe any maximum key sizes for the group prime q; the maximum size of the basic prime p is 512 bits. A popular subgroup size is 160 bits for group prime q, used in the US Digital Signature Algorithm, for example, with basic prime size p varying from 512 to 1024 bits.

Elliptic Curve systems

In Elliptic Curve (EC) discrete logarithm systems, the group structure is based 'on the points on an elliptic curve' (think of a curve in a field). Again, the size of group q is a prime number and the size of group prime q generates the key size of the EC.

Wassenaar Arrangement: The maximum EC key size allowed by the Wassenaar Arrangement is 112 bits. A popular EC key size is 160 bits.

The Other Year 2000 Problem - Root CA Expiration

by Ian Curry,
Vice President,
Entrust.net, Entrust Technologies

The popular media coverage of Year 2000 issues has served as a wake-up call for organizations to make their customers feel confident that their personal information is safe as they conduct business in cyberspace. One-third of European companies now use e-commerce in business procurement, logistics, finance and product development[1]. Further, at least 90 percent of companies expect to use e-commerce for sales and marketing, with 83 percent predicting that they will use it for business procurement[2]. It is estimated that Western Europe's Internet market will be worth US$430 billion by 2003, with 170 million users[3]. With an increase in critical information and transactions being conducted over the Internet, security and the other Year 2000 problem – the Year 2000 Root Certification Authority (CA) expiration – is an important issue to understand. Although not related to the infamous Y2K problem, its potential impact on e-business in early January 2000 represents a significant business issue.

Beginning on January 1, 2000, Root CAs belonging to AT&T Certificate Services, GTE CyberTrust®, and VeriSign™, Inc are set to expire in all Netscape browsers up to release 4.05 and Microsoft's Internet Explorer 3.x. After the Root CAs expire, Netscape 4.05 (and earlier) users will see an important warning message each time they attempt to establish a secure connection to Web servers using certificates from an expired CA (Microsoft IE 3.x does not show such a warning message). Root CA expiration represents a challenging problem to e-businesses because the expiring Root CAs are embedded in tens of millions of browsers. In addition, because the expiration coincides with Y2K, the average user could easily mistake the warning message to be a problem with the Web site and not with a Root CA in their browser. From an e-business marketing and operational perspective, the problem could lead to lost revenue, customer dissatisfaction, and costly technical support and customer service calls.

In the context of Web security, Root CAs are the Certification Authority certificates embedded in (or shipped with) the browser software. Browsers worldwide automatically trust Web server certificates issued by these CAs. Each Root CA actually consists of two keys: a public key and a private key. The private key, which is used to digitally sign certificates, must be held in a secure location and never disclosed. Only the public key, which is used to verify the CA's signature on certificates, is shipped with the browser software. Root CA public keys are stored in certificates. Like all certificates, these Root CA certificates have expiration dates.

After any certificate expires, the public key in the certificate should not be used. Given that the expiring Root CA certificates are embedded in millions of users' browsers, there is simply no easy way for these CAs to solve their expiration problem.

Understanding the problem

Netscape browsers display the following message when attempting to verify a certificate using an expired Root CA. This type of warning message is correct behavior for a security application when it encounters an expired CA.

When Netscape's browsers encounter certificates signed by expired Root CAs, the browsers react correctly by displaying this important warning message to users. These types of warning messages represent the ultimate protection of users from fraudulent activity on the Internet. Consequently, no reputable organization would encourage or train users to ignore these types of warnings.

E-business represents an essential and increasingly important function in many organizations worldwide, so the potential implications of Root CA expiration are important for Web site owners using the Secure Sockets Layer (SSL) protocol to clearly understand and manage the problem with their customers. E-business Web site owners have already expressed a number of concerns about the potential implications of the Root CA expiration problem, including:

- The potential loss of e-business revenue and, perhaps more importantly, customer confidence by those who do not understand the Root CA expiration warning message and believe that the Web site has a serious problem;
- The costs of training help desk support staff to handle telephone calls from concerned users, and the costs of handling those calls themselves (particularly in early January 2000);
- The potential negative ramifications and secondary economic effects to other organizations and partners in the supply and distribution chain of the e-business; and
- The potential to be perceived as having a Year 2000 problem because users will only begin to see the Root CA expiration warning message on and after January 1, 2000.

At the same time, the most significant issue for Web site owners using certificates from expiring CAs is that users must take action themselves to solve the problem. The warning message will continue to appear until the user takes steps to correct the problem.

A recent article in Information Week stated that approximately 20% of all browsers in use on the Internet are Netscape version 4.05 and earlier[4]. With approximately 160 million users on the Internet[5], 20% translates into approxi-mately 32 million users. E-business Web sites should examine their own visiting browser statistics to determine the significance of Root CA expiration on their users. One top e-business Internet site informed Entrust® Technologies that Root CA expiration will affect over 40% of its customers.

The solution

Root CAs are an essential and integral part of browser security, so there are no automated or transparent mechanisms to easily update Root CAs in users' browsers – such a mechanism could represent a serious security hole that could be used by hackers on the Internet. Web sites concerned about Root CA expiration do have a useful way to resolve the problem for the vast majority of their visitors – without requiring visitors to obtain new browsers or download a newer Root CA. The solution is to obtain a Web server certificate from a CA service provider whose Root CA does not expire on December 31, 1999.

E-business Web site owners are concerned about potentially losing revenue and customer confidence because of this problem. They are also concerned about the possibility of handling an influx of help-desk calls, potential negative ramifications on their supply and distribution chains, and the incorrect perception of having a Y2K problem.

The best way for the e-business Web site owners to safeguard against potential backlash from customers, is for to understand the problem and develop a strategy to manage the problem. The Year 2000 Root CA problem can be resolved quickly and easily (without impacting the majority of Web site visitors) by selecting a CA service provider whose Root CA does not expire on December 31, 1999.

About the author

Ian Curry is the Vice President of Entrust.net; an Entrust Technologies business unit focused on Web e-business solutions. Mr. Curry has an MBA from the Massachusetts Institute of Technology (MIT) and an honors degree in Computer Science from the University of Western Ontario. Ian is one of the original employees of Entrust Technologies (he joined the company in 1994) and a frequent speaker at PKI and Internet e-business conferences.

[1] Andersen Consulting survey, September 9, 1999 – www.ac.com

[2] Andersen Consulting survey, September 9, 1999 – www.ac.com

[3] Andersen Consulting survey, September 9, 1999 – www.ac.com

[4] The Year 2000 Certificate Problem, Jason Levitt – Information Week Online, July 5, 1999

[5] Statistics from Nua Internet Surveys. HYPERLINK www.nua.ie – March 1999

Mobile Code - Friend or Enemy?

by Mark Vandenwauver,
Robert Maier,
Joris Claessens,
Calin Vaduva,
ESAT/COSIC, UT Cluj

Since the explosion of the Internet more and more companies go on-line. Security conscious enterprises try to run a secure environment. Tools such as Firewalls and Intrusion Detection Systems have been put in place to make systems as secure as possible. Still, due to the evolution in the interconnection of open systems, and the appearance of new paradigms such as mobile code, even the most modern systems that are nowadays available, cannot stop attacks such as those we describe in this article.

Introduction

Any system connected to the Internet has to deal with two kinds of attacks: attacks that come from the inside (Intranet) and attacks that come from the Internet.

The attacks that come from the inside are most of the time initiated by the users of the system and can be instigated by different motivations. These users have the advantage of being connected to the system and having a certain degree of access rights. They do not need to hack their way into the system. Perhaps this is the reason why most attacks come from the inside (recent informal contacts with NSA suggest up to 70%).

The attacks, coming from the outside, are initiated by hackers. They need to get access to the system before they can move on to compromise it. Any networked system or even a simple computer that is connected to the Internet is vulnerable to these attacks.

Mobile code is code that originates from a machine, called a server, and is executed on a different machine, called a client. Its mobility makes it very useful for various applications. But some of this code might be written by potential attackers and can have all kinds of malicious behavior. The same tools that are useful in the hands of a regular user can be malicious in the hands of a potential attacker.

Mobile code can be embedded into Web pages and executed whenever the page is viewed. Most of the time this is done without the Web surfer having to be aware of this. Mobile code achieves to be platform independent. This could thus lead, at least theoretically, to a platform independent virus. Mobile code mostly travels using the World Wide Web and gets executed within Web browsers such as Netscape Navigator and Microsoft Internet Explorer. It can be filtered out and stopped from executing but this leads to a severe loss of functionality. The tools that are mostly used to create mobile code are Java and Javascript.

This article presents attacks that combine the two kinds of previously mentioned attacks. Attacks mounted using mobile code usually originate from the outside, get passed the filters put in place by the firewalls and work from the inside on the system. These attacks are thus internal attacks originating from the outside. This makes our attacks extremely powerful.

Most of the attacks that were mounted using mobile code were based on flaws [3]. They were trying (most of the times successfully) to exploit flaws in the implementation such as the Java virtual machine and its sandbox model. The attacks that are described herein do not exploit any flaws. They exploit the extra functionality offered by the tools used to create and deploy mobile code. To some extent, they also use social engineering techniques.

Firewalls and IDSs

In order to detect and defeat internal or external attacks, several tools have been put into place.

For detecting internal activities that could lead to a system compromise, Intrusion Detection Systems are available. They monitor the activities that take place inside the system and report (or stop) any suspicious activity.

For stopping external attackers, a system can be equipped with a firewall system. The firewall tracks all the incoming and outgoing traffic and should stop any unwanted communication between the inside and the outside.

An ideal firewall would stop all the attempts coming from the outside while an ideal IDS would detect all the possible attempts coming from the inside. Of course both firewalls and IDSs are subject to policies. They have to allow certain activities to take place. Sometimes it is also difficult to tell whether some activity is malicious or not. For example, how can you tell if a CPU usage of 90% is an attack that tries to put your system down or just a useful process that requires a lot of processing time?

Malicious mobile code

With respect to malicious mobile code, one mostly refers to exploiting bugs in the virtual machine implementation of browsers [3]. These bugs and the exploits have already proven to be a major security concern. In this article, another mobile code attack is presented, which works even if the implementation is perfect. The following two paragraphs discuss the attacks that we implemented using Javascript and Java respectively.

1. Javascript

Javascript is one of the tools used to produce and deploy mobile code. The code produced by Javascript can be embedded within a Web page. It travels across the network within that Web page and gets executed on the client machine whenever the page is accessed. It can also be attached and/or embedded into a Web-based E-mail message and executed when the E-mail is read using a Web based mail reader, such as Netscape Mail or Microsoft Outlook Express. The first possibility leads to passive attacks, the latter to active attacks.

Most of the attacks that can be mounted using Javascript seem harmless but they can disrupt normal operations or gather specific information on the system. This information can then be used in a more elaborate attack.

Javascript code can also be combined with other kinds of mobile code, like Java, to overcome some deficiencies.

Two kinds of attacks we can mount using Javascript are:

Denial of service (DoS) attacks

DoS attacks are most improperly handled by Java and Javascript security enforcing mechanisms. This is because they do not consider DoS attacks as a big threat and also because it is extremely difficult to detect such an attack. While DoS attacks on themselves may seem harmless, they can be used by an attacker to allow him to execute more pervasive types of attacks. For example an IP-spoofing attack requires the legitimate computer to be off-line so it can not respond to the requests.

There are two kinds of DoS attacks: those that crash the system they target and those that abuse its resources, either for their own agenda (cracking passwords for example) or just for slowing the system down. The first ones are usually easy to recover from, just by rebooting the system. The latter are most of the time more difficult to detect, but the recovery from them is equally easy (killing the program). Each attack that was mounted is capable of crashing the browser or the mail reader in which the code is run. For example an attack is able to take control of some of the browser's behavior such that the browser never leaves a page. Whenever the victim tries to leave the *sticky* page, the browser displays it again. Any legitimate process repeated an infinite number of times also leads to a browser or mail reader crash. There are cases in which the infinite sequence of actions can be stopped from within the browser but some

actions like scrolling the contents of a Web page up and down indefinitely make up a browser crashing attack. Some of them are capable of crashing the system. For example an attack directed to the window manager could compromise the entire system by opening an infinite number of windows. Of course there are window managers that are not compromised by this attack but most of them are. These attacks do not need a cover-up. They strike whenever needed by the attacker.

Informational attacks

These attacks need something to lure the user into believing that something useful is being done. For example a screensaver is being installed or some themes are downloaded. In the mean time, the script carries on with whatever it has to do. Some of the attacks in this category dig for information and then send it out to the attacker. The IP address of the client, the browser type and version even the type and version of the operating system represent information that can be found and sent out to the attacker. These attacks do not require any privileges.

Others can do more than that. They use some social engineering techniques and lure the user into granting some rights to the script. For example in one attack, the *screensaver attack*, only the permission to modify the browser in some way is needed. Then it fools the user into believing that a screen-saver has been accidentally activated. The attack emulates the behavior of the screen-saver, blanking the screen and popping up a window after an inactivity timeout. The username and password are asked and when obtained they are sent out to the attacker and the screen is restored. Thus, using some social engineering techniques, the attack is able to obtain the username and password of some users.

2. Java

Java technology gives the possibility to develop mobile code that can be executed on different types of platforms (platform independence), for example in the frame of a browser. In that case the code is downloaded from the server and is then executed on the client system. A possible drawback of this approach is the security aspect. Such Java code may launch different malicious operations against the client system. In order to solve this security problem a Java security mechanism in four levels was developed and implemented: language level, bytecode verifier, Class Loader and Security Manager [2].

Based on the Security Manager component, there are different Java security models currently adopted by major software vendors:

- JDK Java security models (JDK1.0.2, JDK1.1, JDK1.2) developed by Sun;
- Netscape Java security model (Netscape Communications) – based on Netscape's Object Signing Technology [4]; and
- Internet Explorer Java security model (Microsoft) – based on Microsoft's Authenticode Technology.

The JDK1.0.2 Java security model restricts the operations that an applet downloaded from the Internet can execute. Thus an applet is restricted to read/write files from/to the client machine, start execution of a program on the client machine, connect to other machines than the server, etc. This kind of restrictions considerably limits the category of applications for which an applet may be used. The JDK1.1 Java security model solves the limitation of applets using signed applets. A signed applet has all the rights that a local application has. The JDK1.2 Java security model uses also signed applets to implement the security model. Compared with JDK1.1 it introduces the concept of granularity. So a signed applet has only a few special rights, not all of them.

The Netscape Java security model is similar to the JDK1.2 security model but is implemented using a different technology. It also uses a role based security scheme in order to simplify the procedure to allocate some combinations of permissions. In order to obtain special rights to access some private resources an applet should use the Netscape Capabilities Classes.

The Internet Explorer Java security model is similar with the JDK1.1 model, but uses a different technology. A signed applet has permission to access all resources of the client machine.

The Java based attack we present is **not** based on any flaw in the implementation of the Java security mechanism. The attack we describe is related to social engineering. The applet offers some attractive functionality that lures the user to grant some special permissions to the applet. These permissions are then used by our *malicious* code in order to attack the client machine. At this moment the attack is available only for Windows platforms (Win95, Win98, WinNT) and Netscape Communicator/Navigator (versions higher than 4.0) and Internet Explorer (versions higher than 4.0).

The following steps are used in all of our attacks (see also Figure 1):

- Detect the type of the client platform and the browser using Javascript. Adapt the attack for the user platform and for the user browser;
- Present the user with some facilities to justify the request for permissions that we need in order to proceed with the attack. In this idea we have explored some paths: installation of the Microsoft themes, installing different programmes on the client without their interaction, etc. Once the permissions granted we proceed with our installation procedure and with the attack;
- Download themes or executable programs from the server on the client host and start the installation procedure;
- Collect private information from the client system. As an example we have implemented the retrieval of the structure of the file system (directories, files, how they are organised), some configuration files that may contain private information (eg Netscape's configuration file prefs.js);
- Write some information on the client system; and
- Send back the information collected, using different modalities (CGI, email server) that are allowed by the installed firewall system.

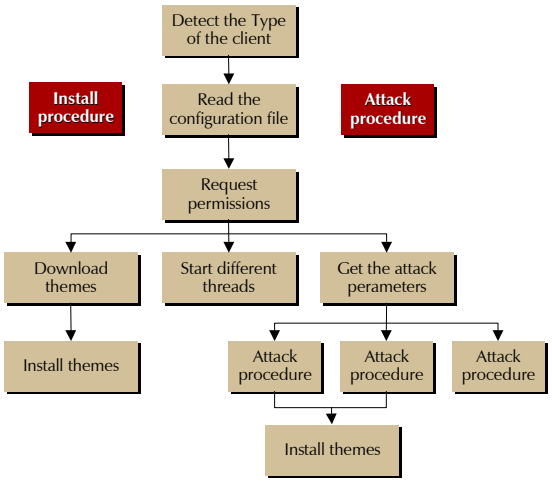


Figure 1: Structure of the attack

We finish with some characteristics of the applet:

- Works on Win95, Win98, WinNT;
- Works for both Netscape and IE;
- Compatible with both Netscape Object Signing technology and with Microsoft Authenticode technology;
- Easy to configure some parameters of the attack: the depth of the file system scanning, the name of the files we are interested in, the modality to sent out information, where to send the collected information;
- Easily add new themes or set-up programmes using a configuration file on the server; and
- Multithreading: there are different threads for installing themes, collect system file info, collect files.

Interested readers can test the attack at [1].

Conclusions

Firewalls and Intrusion Detection Systems are excellent tools to secure an Intranet. They are needed to protect the valuable information that is essential to the future of the enterprise. However, they are sometimes seen as the ideal solutions. Perfect security is not possible. Moreover, the TACK project shows that it remains possible to access the Intranet and obtain valuable information, even if Firewalls and IDSs are installed and properly configured. The reason for this success is twofold: users are easy to manipulate and deceive, and Java and Javascript provide the attackers with a set of tools previously unknown.

To counter these potential threats it is important that the issue of trust is centralized within enterprises. The installation and maintenance of web browsers cannot be attributed to the end user. Moreover the only current real solution to prevent our attacks is to disallow the use of Java and Javascript. Finally, the Java security model in browsers should allow more granularity (such as giving the applet access to a file or directory instead of the whole file system) with respect to requesting extra privileges.

References

[1] The 'TACK' web site, www.cosic.esat.kuleuven.ac.be/tack/

[2] G. McGraw and E. Felten. Securing JAVA: Getting Down to Business with Mobile Code. Wiley and Sons, 1999.

[3] Hostile Applets Home Page, www.rstcorp.com/hostile-applets/

[4] Netscape Object Signing and Capabilities, http://developer.netscape.com/docs/manuals/signedobj

A Practical Approach to Managing Legal Risk in Electronic Business

by Mark Lewis,
Arnheim Tite & Lewis

Think about the basic legal issues you need to address in putting together a "simple" online trade. Take as an example a business-to-consumer electronic trade, where you, the seller, and your customer are in the same jurisdiction. Are you allowed to trade in the goods or services at all? If you are, do you need regulatory approval beforehand? Are you allowed to advertise this online trade? Have you structured the trade to recognise local contract formation rules, like invitations to treat, offers and acceptance? Have you effectively incorporated in your online contract all the terms of the trade and managed your legal risk? What law is there in your home country that could override your terms and grant your online customers rights and greater redress than you had ever intended?

The fact is that, for all the "simple" kinds of online trading, there are many more complex electronic commerce models in which even small and medium-sized businesses participate. There are business-to-business online trades, often involving a host of trading partners. By definition, electronic business transcends national borders. You will have often have to think about a number of complex legal issues and risks in a number of jurisdictions. And then, of course, you have to manage them.

The purpose of this article is to suggest that, given the complexity of online trading, the most useful tool in managing the legal risks in electronic business is an approach or methodology that helps:

- To identify the main legal risks in electronic business, wherever it happens in the world and whether that business is directed to other businesses or consumers; and
- To decide on the most effective ways of managing those risks.

There are three kinds of risk: systemic risk, service dependency/liability risk and regulatory risk.

Systemic risk

Legal barriers to electronic commerce

This is the risk that legal systems do not recognise, or create uncertainty in, online traders' legal rights and responsibilities. All legal systems – even those like the USA and Singapore that have specifically recognised electronic commerce in their laws – suffer from systemic risk. These are sometimes identified as the legal

barriers to electronic trade. The most common of such barriers are:

- The need for some transactions to be in writing, signed or physically delivered in some way (e.g. under English law deeds, marine insurance contracts, guarantees, bills of exchange and real estate contracts);
- The extent to which electronic data can be used in court as evidence of the trade and the legal requirements for audit of electronic trades to make that evidence admissible and/or more compelling;
- That, in most countries, electronic contract formation is not yet specifically recognised;
- That, almost everywhere in the world, digital signatures are not yet recognised, so business cannot safely rely on certification authorities or trusted third parties offering confidentiality services;
- That, in most countries, electronic invoicing and electronic payments are not always specifically recognised. This is especially galling where parties have created secure electronic trading systems, only to find that the revenue authorities in certain countries still insist on paper invoicing;
- That no legal systems or international treaties have yet fully adapted existing intellectual property right protection to digitised products and services or have developed new digital rights. This means that there is a real risk that intellectual property rights in online trading cannot be enforced effectively;
- That there is no internationally agreed way of resolving cost effectively (or at all) disputes arising from online trades; and
- That there are no internationally agreed rules and procedures for determining which legal system or systems govern online trade and whose courts will have jurisdiction to hear disputes that have to be determined in that way.

Managing the systemic risk

This kind of risk is the most difficult to manage, because only governments, legislative or judicial authorities make law. So we cannot ourselves remove these legal barriers to electronic trade.

But some of these risks can be overcome or worked around, such as where a party to a trade or a number of you write your own contractual rules to determine how and when binding contracts will be formed and electronic messages are received, how disputes will be resolved, the rules for the admission in evidence of the trades and whose law will govern the trades. This is common in electronic data interchange agreements: see, for example, the principles set out in the European Commission Recommendation of 19 October 1994 relating to the legal aspects of electronic data interchange (94/820/EC), Official Journal 1994 L 338 and the unofficial May 1995 draft of the UNCITRAL Model Law on Legal Aspects of Electronic Data Interchange (EDI) and Related Means of Communication – www.tufts.edu/departments/fletcher/multi/texts/uni.txt .

Equally, you must accept that some of these risks cannot be overcome or worked around.

The right approach is:

- To identify and assess the impact of all the legal barriers to your online trade;
- Specifically to identify those that you have little real prospect of overcoming;
- To structure your project/trade to overcome or work around the barriers where you can, and to minimise the legal risk where you cannot;
- To try wherever possible to develop your own legal system or rules of your online trading "club". The EDI interchange agreement model is a good one; and
- To identify and, if cost effective, use technology to help manage certain legal risks, eg digital watermarking or "pay as you go" coding in digitised products to overcome the difficulties of enforcing your intellectual property rights in those products.

A number of countries have now put in place, or are in the process of putting in place, legislation to overcome some or all of the legal barriers.

Legislative solutions?

A number of countries have now put in place, or are in the process of putting in place, legislation to overcome some or all of the legal barriers outlined above. In the UK, we await the outcome of the process initiated in early March by the government to Building Confidence in Electronic Commerce – a Consultation Document (URN 99/642). We are expecting a Bill to be introduced in Parliament in late Spring,

but for enactment of the resulting law in the next Session of Parliament.

At the level of the European Union, there are a number of ongoing legislative proposals aimed at facilitating and regulating electronic commerce, including in the areas of transparency and harmonisation of technical standards for electronic commerce, digital signatures, encryption, digital copyright, convergence and governance. And there is, of course, the EU's Proposal for a Directive on Certain Legal Aspects of Electronic Commerce in the Internal Market. Despite political difficulties, it is possible that a directive will be agreed some time in 1999. If so, EU Member States will need to implement it within one year, ie in 2000.

It is too early to say whether these solutions will provide satisfactory solutions to any of the legal barriers. What is clear is that governments are unable in one fell swoop to provide solutions across the board. And there remains the challenge that, to be really effective in making a level playing field for electronic business around the world, there will have to be multi-lateral treaties between governments and transnational organisations. These are some way off.

The service dependency/ liability risk

This is the risk that online trading parties accept, or have imposed on you, greater responsibility, hence liability, than is warranted.

The point here is that, having identified these risks, it is largely up to you to regulate them by contract.

What are the specific service dependency/liability risks?

Overall, they stem from the failure to recognise that there can be – and frequently are – risks inherent in electronic commerce that are not always inherent in physical trading. Just think about the communications links stretched out between the online supplier and customer. From the moment that an electronic message leaves your own network for another, neither you nor your trading partner somewhere else in the UK or around the world is likely to have any control over it, as the message is carried over or under land and sea or in space. And there are few, if any, legal rules to govern the conditions under which you can argue that you should not be held liable for events outside your control, such as non-delivery or late delivery of the message. That is a different proposition to that which applies in the case of physical trade.

Even where these are stretched out between buyer and seller by, for example, the need to carry the goods by sea, there is a venerable and comprehensive law merchant to fill the gaps in commercial relationships.

In this context, there is clearly a legal risk in online trading parties accepting:

- Absolute, rather than reasonable, efforts to perform your respective contractual obligations; and
- Strict, as opposed to reasonable, time limits to perform those obligations.

Equally, recognising the service dependencies, there is a legal risk in failing to allocate risk fairly between traders and even more of a risk in failing effectively to contain your liability in online trade.

Managing the service dependency/ liability risk

The most effective approach here will involve:

- Assessing external legal controls on the management of your risk, e.g. consumer protection laws such as the UK Unfair Terms in Consumer Contracts Regulations 1994 and business-to-consumer as well as business-to-business protection in the UK Unfair Contract Terms Act 1977, which make unenforceable certain exclusions and limitations of liability;
- Delimiting contractual performance according to what is achievable, as suggested above, by moving away from absolute to reasonable efforts obligations;
- Avoid loading the commercial and legal risk according to the value of the transaction. This is a common problem in electronic banking and payments systems, as well as in internet trade. It is always tempting to try to impose greater responsibility, hence liability, for non-delivery or late delivery of a huge payment or very important message. Ultimately, in internet trade, the service dependencies (see above) remain largely the same. So the risk of your payment disappearing into a black hole in cyberspace is likely to be the same for £100 as it is for £1,000,000; and
- In general, putting in place the right contractual framework and, in particular, clear terms of engagement.

Clear terms of engagement

It is worth saying something more about these. The purpose of having such terms is to allocate

risk between you and your online trading partners. There is nothing new in this in commercial transactions. But the challenge, given that many online trades are one-off or at best sporadic, is to incorporate the terms that allocate risk effectively, giving your trading party enough notice of them and, with that notice, the choice whether or not to proceed to trade online with you.

There are at least two main ways of allocating risk.

The first is by stating in your contractual terms your respective service dependencies. Firstly there are those affecting all online trading partners equally, like network outage over which none of you has any control. You can consider specific *force majeure*-type provisions here. Next, you need to contract on the basis of your trading partners managing their own risk where they can, for example, by tracking the delivery route of important messages to detect any failures or errors and by having alternative ways of delivering messages or instructions to mitigate the effects of non-delivery or other misdelivery.

The second is by limiting your liability for direct and consequential losses. (For the purposes of this article, examples of "consequential losses" are loss of business, goodwill and profit, loss of anticipated savings and other contractual benefits, losses caused by third party claims and loss of data.) It is often a difficult balancing act to exclude and/or limit your own liability in a way that manages your risk effectively while being legally enforceable under provisions like those referred to in section (Managing the service dependency/liability risk) above.

Unfortunately, there are no absolute rules, except in the case of business-to-consumer sales. There is little you can do here, but delimit your obligations as far as you can and limit your liability as low as you can.

There is more scope in business-to-business online trade, but your exclusions and limitations will, under UK law, have to satisfy the requirements of reasonableness. There are guidelines in the Unfair Contract Terms Act 1977, but a good rule of thumb is to ask: what am I doing to make things right, if I fail to meet my contractual obligations? The odds are that, if you are excluding as much of your liability as you can and you are limiting the rest without offering anything meaningful to put things right, you may be going too far and your efforts to minimise your risk will be unenforceable. It has become standard information and communications industry practice to exclude all liability for consequential loss. This may seem like a very good idea in the context of electronic business, but beware: just because this loss is indirect or consequential does not mean that a court would uphold your understandable efforts to exclude it totally. (On the contrary, there are signs that the UK courts are becoming less tolerant of such blanket exclusions.)

You must seek legal advice in this area.

The Regulatory Risk

This is the risk that laws in place or to be enacted will either prevent or severely restrict electronic trade. This risk differs from systemic risk, in that the laws here are specifically aimed at electronic business or other electronic transactions (eg data protection laws that severely restrict the cross-border transfer of personal data) or that their effect is more perilous for online traders because of the nature of the online trade (eg being held under such laws to have made investment advertisements to UK residents or to have offered securities to residents in countries in which making such an offer, from wherever it emanates, is unlawful).

You will underestimate the effect of those laws or the time it can take to manage them.

There is also an accompanying risk that you will underestimate the effect of those laws or the time it can take to manage them.

The main regulatory risks

The specific risks arise from the following:

- Competition/anti-trust rules that prevent two or more parties, often in the same sector, from joint venturing to create electronic trading platforms, even if access will be granted to other players in the sector. This writer has seen at least two major electronic business initiatives fall at the first hurdle because of restrictions under EU competition law, in particular Articles 85 and 86 of the Treaty of Rome;
- Data protection rules, such as those now in force throughout the EU, that impose severe controls on the collection, storage, processing and distribution of personal data. These rules have a direct impact on a number of online trades, especially where there is a need to send out of the EEA personal data for batch processing;

- Consumer protection laws, as outlined above. There is likely to be greater regulation, as the EU's *Proposal for a Directive on Certain Legal Aspects of Electronic Commerce in the Internal Market* is designed to offer still greater protection to online consumer customers;
- Telecommunications and broadcasting regulation, which, in the age of convergence, has computer systems, cable TV, voice telephony and Asymmetrical Digital Subscriber Line (ADSL) technology coming together to provide video and multimedia services using telecommunications networks. Those who provide those networks and their trading partners may be subject to telecommunications and broadcasting controls;
- Laws limiting the use of powerful encryption tools; although there are signs that, at least in North America, Europe and some of the more developed Asian countries, law enforcement agencies may be willing to compromise. However, the debate is far from over;
- Specific industry sector regulation, eg in the banking and financial services and pharmaceuticals industries. Under the UK Financial Services Act 1986 and subordinate legislation made under it, there are restrictions on carrying out investment business and investment advertisements. Under the UK Public Offers of Securities Regulations 1995, in connection with offers of listed securities to the public there needs to be a prospectus available in UK. The point here is that electronic business is beginning to penetrate all industrial sectors, so the way in which sectors react will be determined by their respective regulatory régimes; and
- Internet service provider/content provider/website host/trading partner liability. So far, there is little law around the world providing for specific liabilities aimed at online service providers. But there are signs that this is changing, including in the development of case law. In the UK, one of the first (if not the first) specific laws to provide for such liability is the Defamation Act 1996. There is now liability for electronic dissemination, and while there is strict liability for those who fall within the definition of author, editor or publisher, there is also a defence of, amongst other things, reasonable care and lack of actual or constructive knowledge of involvement in the electronic publication. This Act has just been tested in Laurence Godfrey v Demon Internet Limited (1999) QBD 26/3/99. This is the first reported defamation case in the UK involving the internet. The internet service provider lost.

Managing the regulatory risks

As with systemic risk, you cannot change the law. So it will not always be possible to avoid the effects of regulatory risk. But there is also scope to structure your trade or project to avoid, or at least work around, the risk.

An approach to managing the regulatory risk should, at the very least, involve the following:

- Understand the regulatory requirements and assess their impact on your proposed trade from the outset. There is no sense in spending time, effort and money on a project to find that you cannot launch at all or that your launch date is delayed because of a failure to obtain regulatory approvals in time or (in some cases) at all. Likewise, you may need to restructure your online trade to comply with one or more of the regulatory requirements outlined above;
- It follows that you should allow time to obtain whatever regulatory clearances you need and/or to restructure your trade. A rule of thumb is to allow more time than you can possibly have imagined, particularly if you have to obtain approvals or legal opinions from a number of jurisdictions. Bear in mind that, even if you ultimately obtain clearance, you may have to negotiate with the authorities to do so;
- Where possible, from the outset structure your project or trade to meet regulatory requirements or to be in a position to receive regulatory approval. This will help to manage the expectations of business partners, investors and the market. But, in order to do so, you will need to have legal input from the start; and
- If you find that it is not possible or practicable to overcome regulatory obstacles, ensure that you are fully aware of, and understand, the legal limits of your proposed project or trade, and be prepared to restructure the project or trade to operate up to, but within, the legal limits. Having practical guidance drawn up for the project team helps, especially as the technology or your business drivers may push them in the opposite direction. Also ensure that you maintain ongoing regulatory "health checks". Official practice in (if not the law covering)

internet regulation can change rapidly. And with some areas of the law, especially in competition/anti-trust, an online trade may start out as compliant, but become non-compliant as it develops.

The Establishment v the Cyberpunks: lines drawn on the electronic frontier

Depending on what sector of commerce or industry you are in, you may also be concerned to manage some of the risks inherent in the internet society. The self-styled cyberpunks (who include in their number some very gifted information technologists) see the internet as a free society, in which:

- Intellectual property rights should not be used to stifle creativity or the development of products and services, however much R&D funding has been sunk in the development of the intellectual property concerned;
- Entering computer networks or systems belonging to third parties, without their permission, and making changes to those networks, systems or the data residing in them, should not be seen as a crime or some other misdemeanour;
- The most powerful encryption tools should be freely available to ensure the greatest possible privacy of electronic messages; and
- There should be no censorship of any kind of content on the internet.

This is undoubtedly a simplification of the cyberpunk position, which (to give it credit) is often more sophisticated and better explained than as outlined here. But there are clearly bigger risks to be managed by those of you whose interests could be damaged by the cyberpunk position. And that is another discussion altogether.

About the author

Mark Lewis has specialised in information technology law for about 15 years. He is a founding partner of Arnheim Tite & Lewis, PricewaterhouseCoopers' UK correspondent law firm. As well as leading his firm's Information Technology and Telecommunications Law Group, he is global leader of PricewaterhouseCoopers' E-Business Law Network. He can be contacted at mark.lewis@uk.pwcglobal.com.

The End of Public-Key Cryptography or Does God Play Dices?

by Franck Leprévost

In August 1998, the Nevanlinna Prize was given to Peter W. Shor (AT&T Bell Labs) at the International Congress of Mathematicians in Berlin. This award, which somehow corresponds to a Nobel Prize in computer science, acknowledged results with tremendous political, diplomatic and industrial consequences. In one sentence: if quantum computers exist one day, Shor's results will make all current known public-key cryptographic systems useless.

Throughout this article, classical is understood versus quantum (e.g. a classical algorithm works on a classical computer whereas a quantum algorithm makes use of quantum physic and works on a quantum computer). For more on quantum computing, see [7].

Public-key cryptography

Description

As opposed to secret-key cryptography (see [6], [11]), public-key algorithms require two keys per user. As usual in this field, protagonists are Alice and Bob: Alice (resp. Bob) chooses a secret key xA (resp. xB) and publishes (for instance in a phone-book) a public key yA (resp. yB). Bob encodes his message with yA, and sends the result to Alice. Only Alice can decode and recover the original message with her secret key xA.

Public-key algorithms are based on mathematical problems:

- Integer Factorization Problem (IFP): Rivest-Shamir-Adleman (RSA) and Rabin-Williams;
- Discrete Logarithm Problem (DLP): Digital Signature Algorithm (DSA), key exchange of Diffie-Hellman, encoding method of El Gamal and digital signature of El Gamal, Schnorr and Nyberg-Rueppel; and
- Elliptic Curve Discrete Logarithm Problem (ECDLP): analogous of the algorithms above for elliptic curves.

Security

One way to measure the security of a public-key algorithm could be given in terms of the time-complexity of the best-published algorithm which finds the secret key, given only the public key. Without going into the details, let us say that there are three complexity classes: polynomial, subexponential, and exponential. There exists subexponential algorithms solving IFP, and DLP. However, there is no known classical algorithm which solves ECDLP in polynomial or subexponential time.

As a consequence, elliptic curve cryptosystems offer the highest strength-per-key-bit of any known public-key system. With a 163-bit modulus, an elliptic curve system provides the same level of cryptographic security as DSA or RSA with 1024-bit moduli (note that the team managed by Herman te Riele broke the RSA-155 challenge on August 22, 1999. Hence RSA-512 bits is no longer secure).

Standardization

The future standard IEEE P1363 ([8]) specifies common public-key cryptographic techniques, including mathematical primitives for secret value (key) derivation, public-key encryption, digital signatures, and cryptographic schemes based on these primitives. It also specifies related cryptographic parameters, public keys and private keys. The purpose of this standard is to provide a reference for specifications of a variety of techniques from which applications may select. The P1363 project started as the "Standard for Rivest-Shamir-Adleman, Diffie-Hellman, and Related Public-Key Cryptography" with its first meeting in January 1994. The draft (version 9) passed recently the ballot, and the current draft (version 11) addresses comments of ballot members. Despite the recent "attack" connected to ISO 9796, it is expected that the draft becomes a IEEE-standard in 1999. There is an on-going IEEE-P1363A project which addresses complementary techniques to IEEE-P1363.

For security reasons (see 2.2), a very important item in this draft concerns elliptic curve cryptography (ECC). ECC is also being drafted into work items of several international standardization bodies: the American National Standards Institute (ANSI) ASC X9 (Financial Services): ANSI X9.62 ([1]); ISO/IEC 14888, the OAKLEY Key Determination Protocol of the Internet Engineering Task Force (IETF, [9]), and the ATM Forum Technical Committee's Phase I ATM Security Specification.

Quantum computers and quantum cryptanalysis: does God play dices?

Quantum computers

In very coarse terms, the "atomic" unit of information is called a bit in classical computer science, and quantum bit or qubit in quantum computing. By analogy with a classical bit, whose value is 0 or 1, a qubit is a two-state quantum system, and the basic operations of a quantum computer use quantum mechanics. However, even if no law of nature seems to be an obstruction to their construction, the existence of quantum computers is still very hypothetical: quantum computers should make use of quite stable quantum systems satisfying the two following properties:

1. They interact strongly between each other, in order to transport quickly the logical quantum gates
2. They interact weakly with the rest, in order to minimize the errors

There are currently several experimental proposals for the implementation of quantum computers ([3], [4], [5]). According to [7], the most promising approach so far is a spin-off from the medical technology of nuclear magnetic resonance (NMR). However, none of the proposals have been experimentally realized for more than a few qubits.

Factorization: the starting point

Suppose that we are facing an Integer Factorization Problem. The general idea used to factorize an integer N consists in findings $s \neq \pm t \pmod N$ such that $s^2 \equiv t^2 \pmod N$. In this case,

$$(s + t)(s - t) \equiv 0 \pmod N$$

and s + t (resp. s-t) contains a divisor of N. Thanks Euclid's algorithm, one computes (on a classical computer) in polynomial time GCD(s ± t, N), which is a divisor of N.

Quantum factorization algorithm

The quantum factorization algorithm provides (if it exists) the multiplicative period of a residue x (mod N), which is the least integer $r \geq 1$ such that

$$x^r \equiv 1 \pmod N$$

With some luck, r is even and $x^{r/2} \neq \pm 1 \pmod N$. In this case, the equation

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \pmod N$$

allows us to conclude that $GCD(x^{r/2} \pm 1, N)$ is a divisor of N. In general, after some tries, one obtains such an x.

The advantage of a quantum computer resides in the following fact, proven by Shor: if N has L bits, one can find this period in polynomial time, by exploiting the dimension (equal to 2^{2L}) of the state space of 2L qubits, and taking the Fourier transformation on this space. Because the dimension of the space is exponential in L, one can take the Fourier transformation of a sequence of exponential length. As a result, Shor's quantum factorization algorithm is

polynomial, whereas the current best-known classical factorization algorithm is subexponential (see 2.2). In technical terms, the best-known classical factorization algorithm is the so-called Number Field Sieve, whose complexity is $O(\exp(cL^{1/3}\log^{2/3}L))$. The complexity of Shor's algorithm is dramatically reduced to $O(L^2\log L\log\log L)$.

DLP and ECDLP

Shor's ideas (see [16]) solve the Discrete Logarithm Problem in polynomial time as well. Moreover, Shor told us that his methods could be enlarged to solve the Discrete Logarithm Problem for general Abelian varieties ([18]), what includes elliptic curves as a particular case! For more technical details, I suggest reading Shor's original articles ([16], [17], [18]), or surf on [15] (one may also have a look at the report [14], asked by the French scientific community, and [12]).

Conclusion

Nonetheless, the standardization projects (see 2.3) for public-key cryptography are still relevant: Shor's algorithms do need a powerful quantum computer. According to Shor himself ([18]), the construction of the first quantum coprocessor requires at least ten years, even if the DARPA, which depends on the Pentagon, officially allows five millions dollars per year to this project. Would such a discovery in 10, 40 or 100 years mean the come-back in business of carrier-pigeons? Fortunately, an alternative to public-key cryptography exists, and is ironically provided by quantum physic again: Big corporations and research centers (IBM, DRA, British Telecom, Swiss Telekom, Los Alamos National Lab) have started experimental realizations of quantum cryptography ([2]). But this is another story.

About the author

Franck Leprévost can be contacted at leprevot@math.tu-Berlin.de .

Bibliography

[1] ANSI X9.62: Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm. ASC X9 Secretariat-American Bankers Association (1998)

[2] G. Brassard: Quantum cryptography: a bibliography, SIGACT News 24:3 (1993). A more recent version online: www.iro.umontreal.ca/crepeau/Biblio-QC.html

[3] J.I. Ciriac, P. Zoller: Quantum computations with cold trapped ions, Phys. Rev. Lett. **74**, p. 4091-4094(1995)

[4] D.G. Cory, A. F. Fahmy, T. F. Havel: Ensemble quantum computing by nuclear magnetic resonancespectroscopy, Proc. Nat. Acad. Sci. **94**, p. 1634-1639 (1997)

[5] N.A. Gershenfeld, I. L. Chuang: Bulk spin resonance quantum computation, Science **275**, p. 350-356 (1997)

[6] G.C. Grabow: Life after DES. CCE Quarterly Journal, Issue 1 (Spring 1999)

[7] L.K. Grover: Quantum Computing. The Sciences, July/August 1999, p. 24-30. Online version: www.techweb.com/wire/story/TWB19990820S0012

[8] IEEE: P1363 Draft, Version 11, http://grouper.ieee.org/groups/1363/index.html

[9] IETF: Public-Key Infrastructure X.509 (PKIX), www.ietf.org/html.charters/pkix-charter.htm

[10] B.E. Kane: A silicon-based nuclear spin quantum computer, Nature **393**, p. 133-137 (1998)

[11] F. Leprévost: AES: Round 2. Article for PricewaterhouseCoopers (1999)

[12] F. Leprévost: http://www.math.TU-Berlin.de/leprevot/quantum.html

[13] F. Leprévost: Les standards cryptographiques du XXI-ème siècle : AES et IEEE-P1363. La Gazette des Mathématiciens (To appear, 1999)

[14] F. Leprévost: Peter W. Shor, Prix Nevanlinna 1998. La Gazette des Mathématiciens, Vol. **81** (1999)

[15] P.W. Shor: www.research.att.com/shor/

[16] P.W. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal of Computing **26**, p. 1484-1509 (1997)

[17] P.W. Shor: Quantum Computing, Proceedings of the International Congress of Mathematicians, Berlin, Documenta Mathematica, Journal der Deutschen Mathematiker-Vereinigung (1998)

[18] P. W. Shor: Private communication (1998)

Electronic Identities and Achieving Portable Credentials through Standardization

by Magnus Nyström, Senior Research Engineer, RSA Laboratories.

With the emergence of electronic commerce and online interactions with authorities and organizations such as banks and insurance agencies, the need for users to securely identify themselves becomes imminent. Thus, the requirement for an "electronic identity" (EID) application has been identified. This EID will usually be public-key based and consist of one or several private cryptographic keys together with corresponding certificates. The key will be used to authenticate the owner, and to digitally sign transactions. Since these transactions may well be confidential, sensitive or otherwise in need of protection, the secure storage and access of the private keys becomes an issue of highest importance. Coupled with this is a strong requirement of portability of credentials (the keys and certificates), since citizens and users of these systems will be mobile and wishing to have access to these services at many locations.

The approach to solve this problem is usually to employ some type of cryptographic tokens, such as Integrated Circuit Cards (IC cards or "smart cards"), since they are capable of providing not only portable and secure storage but also a secure computation environment. Further, IC cards allow for a wide range of user credentials such as keys, certificates and passwords. Because of this, it is widely recognized (cf. [2]) that they offer great potential for secure identification of users of information systems and electronic commerce applications. Several countries, among them the Scandinavian ones and Germany is also actively looking into the possibilities and requirements for *national* EID tokens.

Unfortunately, the use of cryptographic tokens for authentication and authorization purposes has been hampered by the lack of interoperability at several levels (cf. [1]). First, the industry has been lacking a standard for storing a common format of digital credentials (keys, certificates, etc.) on them. This has made it difficult to create applications that work with tokens from a variety of technology providers. It has also created a significant problem for end-users since tokens (and credentials) are tied to particular applications running against particular application-programming interfaces.

Second, the limited room on many tokens together with a consumer expectation of universal acceptance will force credential sharing on credential providers. Without agreed-upon standards for such sharing, acceptance and use of the tokens, both by application developers and by consumers, will be limited.

To optimize the benefit to both the industry and end-users, it is important that solutions to these issues be developed in a manner that supports a variety of tokens, operating environments, application programming interfaces, and applications. Only through this approach can the needs of constituencies be supported and the development of credentials-activated applications encouraged.

PKCS #15

One recent such solution, RSA Laboratories' PKCS #15 [7], was developed as a framework to allow token-holders to use their cryptographic tokens to electronically identify themselves to any application regardless of the application's token interface. The standard specifies how personal credentials are to be stored and accessed on tokens. One important design goal was to maintain consistency with existing, related standards (c.f. [3]), while expanding upon them only where necessary and practical. Furthermore, the standard builds on experiences from earlier, related work, eg [1] and [8].

One may ask whether a standardized cryptographic token programming interface (API), such as RSA Laboratories' PKSC #11 [6] or Microsoft's PC/SC suite of specifications ([4]) is not enough. But the answer is that APIs alone can not offer this functionality since an API specification is aimed at offering applications a uniform *interface* to cryptographic tokens. This means that different tokens requires different PKCS #11 implementations, and unless a user's desktop has the "right" PKCS #11 library installed, the user will be unable to use the token on that desktop.

The EID Application profile of the first version of PKCS #15 has been defined as a subset of PKCS #15 suitable in environments where electronic identities are deemed useful or necessary. It has been developed in collaboration with several standardization bodies' working groups and has been adopted for us by eg the WAP forum [5]. PKCS #15 is being considered by several nations planning for *national* electronic identity cards as well.

The EID application may well be the "killer" application the smart card industry has been waiting for, by enabling secure, public key based electronic identification of card holders and service subscribers.

References

[1] DC/SC, "*Interoperability Specification for Digital Certificates - Storage of Digital Certificates on ICCs,*" draft version 0.2, Digital Certificates on Smart Cards Working Group, 1998.

[2] S. Guthery S and T. Jurgensen, *Smart Card Developer's Kit*, Macmillan Technical Publishing, 1998.

[3] ISO/IEC 7816-5, "*Information Technology - Identification cards - Integrated Circuit(s) cards with contacts - Part 5: Numbering systems and registration procedure for application identifiers,*" International Organization for Standardization, 1994.

[4] PC/SC, "*Interoperability Specifications for ICCs and Personal Computer Systems,*" The PC/SC Workgroup, December 1997. More information about PC/SC is available at www.smartcardsys.com.

[5] P. King, "The Wireless Application Protocol (WAP)," *Proceedings of RSA Data Security Conference '99,* San Jose, USA, 1999. More information about the WAP forum is available at www.wapforum.org.

[6] RSA Laboratories, "*PKCS #11: Cryptographic Application Programming Interface,*" version 2.01, December 1997. Available from www.rsa.com/rsalabs/pub/PKCS/html/pkcs-11.html.

[7] RSA Laboratories, "*PKCS #15: Cryptographic Token Information Format Standard,*" version 1.0, April 1999. Available from www.rsa.com/rsalabs/pub/PKCS/html/pkcs-15.html.

[8] SEIS, "*SEIS Cards - Electronic ID Application v2.0,*" The Association for Secured Electronic Information in Society, 1998.

About the author

Magnus Nyström joined RSA Laboratories in 1998. His research interests include smart cards, security architectures and cryptographic protocols. He received his M.S. from the University of Uppsala, Sweden in 1990 and has been working in the field since then.

The Beer Bottle Cipher

by Ron Rivest,
MIT Department of Electrical Engineering and Computer Science

Last week an MIT student hacker broke into the famous Yale University secret drinking society known as "Skull and Bones". He made a startling discovery that has implications for national security, saloons and camp counselors nationwide.

What he discovered gives a surprising explanation for the origin and meaning of the well-known drinking song "99 Bottles of Beer on the Wall." The song, familiar to many, starts with the verse:

> *99 bottles of beer on the wall,*
> *99 bottles of beer.*
> *Take one down,*
> *Pass it around,*
> *98 bottles of beer on the wall.*

Successive verses are the same, with the numbers reduced by one each time. The song ends (sadly, but in glorious harmony) with "No bottles of beer on the wall".

Apparently, this drinking song describes an encryption procedure used by Skull and Bones' members to protect sensitive information. The procedure, called the "Beer Bottle Cipher," was devised in the early 1700's by a mathematically-inclined Skull and Bones member. The song was crafted as a mnemonic for the procedure.

The MIT student discovered a yellowed manuscript in the S&B vault describing the origin and meaning of the song. ("Lock-picking that vault was a piece of cake," the student was reported as saying.)

The Skull and Bones society uses the Beer Bottle Cipher to protect its most valuable information. For example, it protects embarassing personal secrets revealed by new members at their initiation ceremony. (Details of the initiation ceremony, such as whether it is actually held in the nude, as has been reported, were not described in this manuscript.)

The MIT student has anonymously posted a copy of the manuscript on the Net. This note gives a technical overview of the cipher.

This discovery may have implications for the current congressional debate about encryption policy, since current export policy would now prohibit the singing of this song in the presence of foreigners.

(In recognition of this development, the US Navy has just instructed its sailors to begin the song with 56 bottles of beer rather than the conventional 99 bottles of beer when they are in a foreign port, or in the presence of foreigners.

And Louis Freeh is rumored to be asking Congress to pass a constitutional amendment banning the song altogether.)

We now give the encryption procedure itself.

Suppose we start with "n bottles of beer on the wall". Imagine that this row of bottles holds an n-digit number – each bottle holds one decimal digit. (Imagine the bottles lined up left to right, with the left-most bottle holding the most significant digit.)

The plaintext to be encrypted is first represented as a number, using two bottles for each letter (A = 01, B = 02, and so on). A "space" is represented as 00. Thus, the secret "BALD MOTHER" would be represented by the number 02011204001315200805 18, using 22 bottles.

If, as in this case, the plaintext needs fewer than 99 bottles, then it uses just the right-most bottles, and the left-most bottles hold zeros, so the total number of bottles is 99. (For longer secrets, start out with more bottles, and sing more verses.)

There is also an encryption key, known as the "skull". The skull is a long secret number known only to the president and vice-president of the society. (George Bush (senior) is believed to have served as an S&B president, which may help explain his later political successes.)

In addition, there is the "table", which is where the "empties" go.

That is, when you "take one down, pass it around", one bottle is taken off the wall (from the right end) and put down at the right end of the row of empties. In the encryption procedure the bottles on the table are not really empties, since they still contain digits, and the actual procedure is a bit more complicated.

Anyway, you start with n bottles of beer on the wall holding the plaintext and end up when the song is over with n empties on the table holding the ciphertext.

The procedure is complicated enough that you probably should not be drinking beer when you try to do it. The song helps you keep on track throughout.

Once you have got set up to encrypt, with the plaintext on the wall, skull in hand, and table empty, you just sing the song. Each phrase in the song tells you exactly what to do next. The four phrases are:

"*k bottles of beer*"
"*on the wall*"
"*Take one down*"
"*Pass it around*"

Each phrase has a meaning, instructing you how to encrypt, as follows:

"k bottles of beer"

First you take the left-most bottle of beer on the wall and move it over to the right-most end. The k bottles in a row on the wall represent a k-digit number. As you sing "k bottles of beer" you multiply that number by the quantity (10k+1), discarding high-order bottles if necessary.

Example:

number on the wall = 537
sing "3 bottles of beer"
move left-most bottle to right end
new number on wall = 375
multiply by 31 (which is 10*3+1)
result = 11625
new number on wall = 625

"on the wall"

As you sing "on the wall", you add the skull to the number on the wall, keeping only the low-order k bottles.

Example:

number on wall = 625
sing "on the wall"
skull = 79723400746524399876110 87
sum = 79723400746524399876117 12
new number on wall = 712

"Take one down"

As you sing "take one down", you remove the rightmost bottle from the wall. Call the digit in that bottle the "bone". Don't put the bone on the table just yet...

Example:

number on wall = 712
sing "take one down"
new number on wall = 71
bone = 2

"Pass it around"

As you sing "pass it around", you do the following.

Suppose you start with 't' bottles on the table, representing a t-digit number. Define the "big bone" to be a (t+1)-digit number each of whose digits is the bone. Then you add the big bone to the ten times the number on the table, and keep only the low-order (t+1) digits of the result.

Example:

number on table = 587623 (t = 6)
sing "pass it around"
bone = 2
big bone = 2222222
10 x table = 5876230
sum = 8098452 (now t = 7)

The output of the encryption procedure is the number remaining on the table when you are done.

That's the entire encryption procedure – the "Beer Bottle Cipher".

The manuscript didn't give the decryption procedure, but merely advised the president to consult a society member who knew some number theory if he needed to decrypt something. [For the mathematically inclined, the only somewhat subtle part is undoing the "k bottles of beer" operation, which can use a precomputed table of the multiplicative inverses of (10k+1) modulo 10^k, for k = 1, 2, ..., 99.]

The actual security of this cipher seems to be an open question... Can it be broken?

Investigations are now underway concerning the true origins of the song "On the Twelfth Day of Christmas"...

(Thanks to Ian Goldberg and David Wagner for some "beer review" ...)

About the author

Professor Rivest is the Webster Professor of Electrical Engineering and Computer Science in MIT's Department of Electrical Engineering and Computer Science. He is a member of MIT's Laboratory for Computer Science (where he served as an Associate Director until June '99), is a member of the lab's Theory of Computation Group and is a founder of its Cryptography and Information Security Group. He is also a founder of RSA Data Security.

Professor Rivest can be contacted at rivest@theory.lcs.mit.edu .

Management Responsibility for Security

by Thomas Warner Huppuch, Vice President and Corporate Counsel, Fortress Technologies Inc.

Corporate officers and directors may be personally liable for failing to secure information systems. Recent cases and articles show this trend is likely to continue. Management needs to take active measures to address information security before problems arise.

Duty to protect information assets

The primary asset of any company is the information residing on its computer systems. Networks today house customer data, sales information, as well as engineering developments and other trade secrets.

Unless information systems are secure these vital assets can be easily lost or stolen by competitors, contractors and disgruntled employees. Management has a fiduciary duty to protect the assets of the company. Failure to take appropriate measures to safeguard those assets may be grounds for holding officers and directors personally liable.

Record keeping and reporting obligations

Management also has a duty to ensure that adequate records and information systems are maintained to enable the company to satisfy its legal obligations.

Every company is required to maintain financial records sufficient to meet accounting standards. Every publicly traded firm must also satisfy SEC reporting requirements. The failure to maintain these systems sufficient to meet legal requirements can have serious legal consequences for the company and senior management.

In addition to financial records, every company today faces a plethora of rules and regulations mandating records and reports on virtually every aspect of the company operation. A few examples are tax, customs, OSHA, environment, employment, government contracts, FDA, FCC, etc..

Information systems that are not secure are likely to produce inaccurate information. Filing tax returns or submitting erroneous invoices can create serious civil and criminal liability.

Liability for inaccurate information

Ironically as the use and reliance on computer networks has grown so too has the trend toward holding Senior Management liable for inaccurate information. The following is a partial list:

- SEC Act of 1934;
- Truth in Negotiation Act;
- False Claims Act; and
- Internal Revenue Code.

Statistics also show the rapid growth in the number of cases and the amount of fines imposed. Since 1990 the federal government has indicted an average of 400 companies. This includes 10% of the Fortune 500. This is a tenfold increase since the 1980s.

Another notable statistic shows the increase in suits for False Claims. Since 1986 over 2,000 suits have been filed and companies have been forced to pay billions of dollars in fines and punitive damages.

In addition there has been a dramatic increase in finding corporate officers and directors personally liable in both civil and criminal cases. In fact, the Federal Sentencing Guidelines encourage companies to hold senior managers personally responsible or suffer even greater punishment for the company.

Recent cases

In re Caremark International is a 1996 case which states in part that Directors can be found liable for failing in good faith to assure an adequate information and reporting system existed to ensure compliance with all relevant laws. In a 1997 case concerning W.R. Grace, the SEC stated officers and directors have an affirmative responsibility to ensure shareholders receive accurate and complete disclosure of information required under federal securities laws.

Prevention is the key

The US Federal Sentencing Guidelines also show there are major benefits to companies and managers who try to prevent problems. Investing in information security is also an investment in compliance. By ensuring information assets and systems are secure and accurate management is also helping to prevent lawsuits and claims from shareholders and others.

Demonstrating due care

To limit personal and corporate liability for information security Management must show it exercised due care. In this context due care means showing several actions have been taken including:

- Adopting or exceeding industry standards;
- Complying with applicable government rules;
- Participate and review surveys;
- Designating a senior officer or director with authority and resources;
- Adopting written policies; and
- Conducting training and audits periodically.

In summary we can expect to see the trend toward management liability for information system security to continue. The only way to minimize the risk of personal liability is to adopt a proactive security program.

PKI - the Essential Elements for Secure E-Business

by Duncan Reid,
Marketing Manager EMEA,
Entrust Technologies

Bill Gates and Michael Dell last month articulated a truth already widely understood when they prophesied that companies which fail to grasp the challenge of e-business will probably be dead within five years.

Much of industry and commerce already knows this, but is wary of e-business without adequate security. Without security, there can be no trust. Without trust, there can be no e-business.

Microsoft itself was rudely reminded of this just two days before Gates and Dell made their joint announcement. Hackers gained access to user details on Microsoft's Hotmail service, causing the e-mail system to be temporarily shut down while security was revised. Media reports spoke of a subsequent potential lack of confidence in Hotmail causing users to desert to rival services.

Ideally, trust in the context of the networked world should map the same model that we have used for thousands of years in the physical world. Written references or prior knowledge assure us that we know who we are doing business with. Strategies ensure that others not party to a confidential deal cannot see or tamper with paper-based information that is stored and in transit. Pen-on-paper signatures are a sign of our agreement.

Public Key Infrastructure (PKI) emerged five years ago as the first practical way of replicating these crucial physical-world mechanisms across networks. Good PKI is inherently scalable. And, by creating trust, it can be used both as a shield against a wicked world, and as an e-business enabler.

But all PKIs are not equal. The most mature and most widely deployed PKI is now on its fifth version having been launched five years ago. Alternatives, with varying levels of functionality, continue to emerge but it is clear that industry-wide agreement on just what attributes a generic PKI should have is yet to emerge. We could, however, be close.

A PKI satisfies the four key requirements of good security; access control, authentication, data integrity and non-repudiation. It does this by providing end-users with integrated electronic identities (certificates), digital signatures and encryption facilities.

If users can't simply and easily take advantage of the encryption and digital signature facilities offered by the PKI to use their applications in a secure and trusted manner, then the PKI is clearly useless. The most fundamental requirement of any PKI is therefore transparency and ease of use to the end-user.

In addition to user transparency, a PKI needs to provide the following functions:

Figure 1

- Public key certificates;
- Certificate repository/distribution;
- Certificate revocation;
- Key backup and recovery;
- Non-repudiation of digital signatures;
- Timestamping;
- Automatic update of key pairs and certificates;
- Management of key histories;
- Support for cross-certification; and
- Transparent, seamless interfacing with end-user applications.

Public key certificates

For any security scheme to work and encompass large numbers of people doing business electronically, each person must be confident that other parties are who they claim to be. A PKI achieves this by giving each user a registered identity in the form of a digital public key certificate.

These certificates are created and issued by a Certification Authority (CA). The CA could be a operated by a department within the organisation, or by an independent organisation that charges for providing the service (a Trusted Third Party or Trusted Service Provider). (NB: the term CA is commonly used to refer to both the technological solution which creates the certificates as well as the organisations which run the CA technology). As long as users trust a CA and its business policies for issuing and managing certificates, they can trust certificates issued by that CA. This is commonly known as third-party trust.

CAs create certificates for users by digitally signing a set of data that may include, among other information, the user's name, the public key of the user, the validity period of the certificate and whether the public key is to be used for encrypting data, verifying digital signatures, or both. Since the integrity of a certificate can be determined by verifying the CA's signature, certificates are inherently secure and so can be distributed in a completely public manner, for example through publicly-accessible directory systems.

Certificate repositories and certificate distribution

Once a certificate has been issued by a CA, it is stored in a certificate repository (directory) so that it is available for automatic use by applications. The consensus among PKI vendors is that the best technology for certificate repositories is provided by directory systems that are LDAP (Lightweight Directory Access Protocol)-compliant. LDAP systems can be scaled to support very large numbers of users, respond rapidly and efficiently to search requests, and can be located throughout the network to meet the requirements of even the most highly-distributed organisations

Fundamentally too, they can also be used to support certificate revocation.

Certificate revocation

There are several reasons why a certificate may need to be revoked prior to the end of its validity period. For instance, the private signing key or decryption key may have become compromised, the person may have left the organisation, or may have been 'barred' for some other reason – a poor credit rating, or professional misconduct, for example.

The CA must therefore be able to publish (securely) the status of each certificate in the system in the form of a certificate revocation list (CRL), which is conventionally stored in the directory. The end-user's applications should then automatically and transparently check the CRL for the certificate's current status before using it.

Key backup and recovery

An organisation will lose valuable, perhaps mission-critical information, if the decryption keys are lost or damaged and no backup and recovery mechanism exists. Furthermore, without a secure backup and recovery scheme, some end-users may choose not to encrypt their most valuable and sensitive information for fear of losing it. This would fundamentally undermine security.

Decryption key backup and recovery is therefore a commercial imperative, but it needs to be quick, easy and inexpensive to operate.

Non-repudiation

Non-repudiation is a vital function of a trustworthy security environment because it prevents individuals later denying their involvement in transactions. Imagine the scenario if you were able to successfully deny a share or stock trade that you had made but did not like the outcome of! In the paper-world, non-repudiation is achieved through the use of physical signatures. The PKI uses a digital signature to achieve the same end.

Each user in a PKI has a signing key pair in addition to the encryption key pair. Used solely to create digital signatures, the signing keys are generated and securely stored under the sole control of the user at all times. If a signing key is lost, or needs to be replaced, the user simply creates another pair. Signing key pairs are not backed up, indeed to do so would make it impossible for the PKI to support non-repudiation.

Timestamping

Timestamping "stamps" the transaction with the exact time that an exchange or transaction takes place. Together with non-repudiable signatures and automated certificate revocation checking, timestamping creates a core functionality set that is increasingly being labelled **"Notarisation"**.

Management of key pairs

As stated earlier, the PKI must be as transparent as possible to end-users. That means that among other functions, the updating of encryption/ decryption keys pairs should be handled automatically by the PKI, with the history of previous decryption keys being maintained both centrally by the back-up and recovery system, and by the client-side software.

Cross-certification

Two models for certification currently exist:

In the ***centralised model***, certificates are issued, upon payment of a fee, by a third-party certification authority (CA) that in most cases knows nothing about the applicant. Certificates may have a time limit, but practical considerations, including geographic separation, make it difficult, if not impossible, to support industrial-strength operating practices.

In contrast, ***cross-certification*** sees user's organisations or communities of interest issuing certificates to their own employees, suppliers, customers or members, and honouring certificates issued by industry or community peers applying a similar high standard of vetting. Because each issuing authority has direct knowledge of recipients, and can also swiftly revoke certificates, the cross-certified model offers a greatly enhanced level of security assurance.

A PKI supporting cross-certification therefore provides a very flexible method of building the large validated communities of interest that are required for automated business transactions, Web-based business, and electronic commerce. It also replicates the trust models used manual or paper-based transactions since the dawn of commerce.

Fundamental to the operation of the cross-certification model is the ability of the client-side software to verify the trustworthiness of a user certificate signed by a cross-certified CA.

PKI-enabled end-user applications and client-side software

The ultimate value of any PKI is bound to the ability of end-users to use encryption and digital signatures to perform some useful task. The PKI has little value on its own.

A PKI must therefore work consistently and transparently across all the required end-user applications on the desktop – for example, e-mail, Web browsing, e-forms, file/folder encryption – while at the same time supporting all of the generic functions in Figure 1. Applications are increasingly being delivered PKI-ready. Those that are not can be PKI-enabled through the use of freely-available developer toolkits.

In addition, the PKI should enable end-users to encrypt and decrypt information even when they are roaming and disconnected from the infrastructure of the PKI. To maximise usability and minimise cost, any client-side software should support multiple types of key storage devices such as smart cards, PC cards, tokens or secure files. (It should also enable the use of a single key storage device across all PKI-enabled applications, for example, a notebook PCs own hard drive).

Summary

The goal of a PKI is to establish and maintain a trustworthy electronic communications infrastructure. This goal is achieved by providing end-users with easy to use electronic identities (certificates), digital signatures and encryption facilities.

Only a comprehensive and fully-managed PKI can achieve the goal of establishing and maintaining usable trust. A PKI that does not offer all the facilities discussed will prove difficult and costly to use – a "certificate pump" is not enough.

With a PKI providing a core, unified security infrastructure for trusted e-business, multiple new applications, products and services can be rapidly developed and deployed.

About the author

Duncan Reid can be contacted at duncan.reid@entrust.com .

AES: Round 2

by Franck Leprévost

On August 9, 1999, NIST opened the Round 2 (which will continue till May 2000) of the AES competition and announced the five finalists: MARS, RC6, RIJNDAEL, SERPENT and TWOFISH. The technical analysis of the finalists will be presented during the third AES conference in New York on April 13-14, 2000. As soon as one (or several) winner(s) is (are) known, it will be proposed as a FIPS after a further examination period of six to nine months. It is expected that AES will become a FIPS in 2001.

The competition is still widely open, and the performances of the finalists are currently compared (see [1], [2], [4], [5],[6], [8], and of course keep in touch with [7]). For instance, how do AES finalists act on chip cards? How do they react to Differential Power Analysis? These technical aspects are currently addressed.

A strategic aspect was pointed out by European companies and journalists, who independently asked me if AES presented a risk of eavesdropping by U.S. government agencies, principally the National Security Agency (NSA). Clearly, AES was a U.S. Government initiative, initiated by NIST. Despite that, my answer was no: submissions were international. The same holds currently for cryptanalysis (the art of breaking codes) as well! Moreover, NIST depends on the Commerce Department of the U.S. Government, even if there are natural connections between NSA and NIST. Finally, each company is always free to use the cryptosystems it wants, but with the utmost care!

In any case, measures are necessary for companies or organizations which need to rely on secure communications. A general rule is to use algorithms that have been scrutinized by the international cryptographic research community and are included in standards. More concretely, corporations which use DES should move at least to Triple-DES as soon as possible, and prepare to adapt to AES. Those using broken proposals to AES should definitively update their systems. These are necessary – but unfortunately non-sufficient – measures to minimize corporate espionage and enter life after DES ([3]).

References

[1] *cAESar Project:* www.dice.ucl.ac.be/crypto/CAESAR/caesar.html

[2] *B.Gladman:* AES Algorithm Efficiency. www.seven77.demon.co.uk/cryptography_technology/Aes/

[3] *G.C. Grabow:* Life after DES. CCE Quarterly Journal, Issue 1 (Spring 1999)

[4] *L. Granboulan:* Analysis. www.dmi.ens.fr/~granboul/recherche/AES.html

[5] *L. Knudsen, V. Rijmen:* Block Cipher Lounge. www.ii.uib.no/~larsr/aes.html

[6] *H. Lipma:* Efficiency Testing table. http://home.cyber.ee/helger/aes/

[7] *NIST AES Home Page:* http://csrc.nist.gov/encryption/aes/aes_home.htm

[8] *B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson*: Performance comparisons of the AES candidates. www.counterpane.com/AES-performance.html

About the author

Franck Leprévost can be contacted at leprevot@math.tu-Berlin.de .

Upcoming Conferences

November 9-11, 1999

The Second International Conference on Information and Communication Security

Sydney, Australia

Original papers pertaining to all aspects of computer systems and information security are solicited for submission to the Second International Conference on Information and Communication Security (ICICS'99).
www.cit.nepean.uws.edu.au/icics99/cfp.html

November 14-17, 1999

The Computer Security Institute's 26th Annual Computer Security Conference & Exhibition

Marriott Wardman Park, Washington DC

For information on attending, call +1 (415) 905-2626, or send email to csi@mfi.com.
www.securant.com/ie/frameset_events.html

November 15-17, 1999

The 26th Annual Computer Security Conference and Exhibition

Washington, D.C.

The 26th Annual Computer Security Conference and Exhibition is the ultimate conference for information security practitioners, whether you are a seasoned professional or new to the industry. Here's where you'll get the training, contacts, exploration of ideas and practical solutions that will make a real difference in your career.
www.gocsi.com

December 6-10, 1999

15th Annual Computer Security Applications Conference

Phoenix, Arizona

Practical solutions to real security problems
www.acsac.org/

January 16-20, 2000

RSA 2000

San Jose McEnery Convention Center,
San Jose, CA.

The ninth annual RSA Data Security Conference delivers keynote presentations from industry leaders and national policy makers, plus more than 150 individual break-out sessions on topics ranging from the latest in cutting-edge crypto-graphic research to the most current implementations of enterprise security and secure electronic commerce.

www.rsa.com/rsa2000/

February 2-4, 2000

Network and Distributed System Security Symposium

Catamaran Resort Hotel, San Diego, California

This symposium aims to foster information exchange among researchers and practitioners of network and distributed system security services. The intended audience includes those who are interested in practical aspects of network and distributed system security, with the focus on actual system design and implementation, rather than theory.
www.isoc.org/isoc/conferences/ndss/2000/

Feb 24 - Mar 1, 2000

CEBIT 2000

Hannover, Germany

CEBIT trade fair highlights all the latest trends in IT and telecommunications and offers a comprehensive line-up of products
www.cebit.de/index_e.html

March 21-27, 2000

SANS2000: The Ninth International Conference on System Administration, Networking, and Security

Omni Rosen Hotel,Orlando, FL

Co-sponsored by the SANS Institute and SAGE.
www.sans.org/newlook/events/sans2000.htm

April 1-6, 2000

Infosec World 2000

Orlando, Florida

This annual security event will enable you to meet with decision-making infosecurity professionals from high-profile companies and government.
www.misti.com/conference.asp

April 11-13, 2000

Infosecurity Europe 2000

National Hall Olympia, London

A dedicated IT security forum in Europe, bringing together professionals interested in IT security with suppliers of security hardware, software and consultancy services, and aims to broaden awareness of the commercial importance of secure and reliable access to corporate information.
www.infosec.co.uk/page.cfm

May 1-4, 2000

Entrust SecureSummit 2000

Dallas

Entrust SecureSummit 2000 will be held in Dallas, Texas USA with over 2000 estimated attendees. Selected customers, partners, developers and analysts will be presenting their deployment stories, in-depth technical issues and PKI solutions. In conjunction, Solutions Expo will include exhibitors from solutions suppliers, consulting and systems integration organizations, companies with extensive expertise in PKI implementations, and other leading vendors.
http://securesummit2000.entrust.com/

Call for Articles

If you are interested in contributing to this publication, we invite you to submit articles containing your thoughts, ideas and concepts.

Contribution guidelines for papers being submitted to the Cryptographic Centre of Excellence Quarterly Journal are:

- Topic must fall under the umbrella of cryptography, security and/or privacy;
- Articles should not be of a promotional or product marketing nature;
- All submissions will be reviewed for content and may be declined at the discretion of the editor (for example, if the tone and/or content is overtly promotional or product marketing-oriented);
- Maximum article length to be 5,000 words plus tables/graphics;
- Submissions must be original work and, where appropriate, give credit to the original author(s);
- The editor reserves the right to edit the text with the agreement of the author; and
- All submissions must be made in MS Word or .RTF format.

PricewaterhouseCoopers reserves the right to re-format for publication purposes and re-distribute as appropriate.

Authors maintain ownership of all submissions.

Completed submissions or abstracts should be submitted via email to either:

Geoffrey.Grabow@uk.pwcglobal.com

John.Velissarios@uk.pwcglobal.com.