Selecting Cryptographic Key Sizes in Commercial Applications[1]

Arjen K. Lenstra

Eric R. Verheul

1 Introduction

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, e.g. 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;
- 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.

---

[1] This paper appeared in the autumn '99 issue of the PricewaterhouseCoopers Cryptographic Centre of Excellence (CCE) Quarterly Journal.

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, i.e., 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.

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;
- In Section 4 we give further comments on our model.

2 The Model

As soon as any reasonable doubts about the quality of the system's design have been dispelled, i.e. 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;
III. Cryptanalysis: the effectiveness of attacks during the life span of the information.

2.1 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.

2.2 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 $2^{10*12/18} \approx 100$ 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, and 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.

2.3 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, i.e., 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.

3 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 below.

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)	Correspondir no. of Years (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	$1.22 * 10^{19}$	$2.70 * 10^{16}$	$2.22 * 10^9$

4 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, i.e., 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.
- 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 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, i.e. 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 (e.g. 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.

5 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 (e.g. 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.

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.

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.

6 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, e.g. 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.

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, i.e., 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);
- 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;
- 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;
- 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 2^{512}. 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.