A New Public-Key Cryptosystem

David Naccache
Gemplus Card International
1 place de la Méditerranée
Sarcelles CEDEX, F-95206, France
`100142.3240@compuserve.com`

Jacques Stern
Ecole Normale Supérieure
45 rue d'Ulm
Paris CEDEX 5, F-75230, France
`jacques.stern@ens.fr`

Abstract. This paper describes a new public-key cryptosystem where the ciphertext is obtained by multiplying the public-keys indexed by the message bits and the cleartext is recovered by factoring the ciphertext raised to a secret power. Encryption requires four multiplications / byte and decryption is roughly equivalent to the generation of an RSA signature.

1 Introduction

It is striking to observe that two decades after the discovery of public-key cryptography, the cryptographer's toolbox contains only a dozen of asymmetric encryption schemes. This rarity and the fact that today's most popular schemes had so far defied all complexity classification attempts strongly motivates the design of new asymmetric cryptosystems.

Interestingly, the cryptographic community has been relatively more successful in the related field of identification, where a user attempts to convince another entity of his identity by means of an on-line communication. For example, there have been several attempts to build identification protocols based on simple operations (see [19, 21, 22, 16]). Although the devising of new public key cryptosystems appears much more difficult (since it deals with trapdoor functions rather than simple one-way functions) we feel that research in this direction is still in order : simple yet efficient constructions may have been overlooked and, in a way, the present paper is an example of such a situation.

As observed by [18], most asymmetric encryption schemes present the following common design morphology :

• Start with an intractable problem P and find an easy instance $P[\text{easy}] \in P$ which should be solvable in polynomial space and time.

• Shuffle or scramble $P[\text{easy}]$ until the resulting problem $P[\text{shuffle}]$ does not resemble $P[\text{easy}]$ any more and becomes indistinguishable from P.

• Publish $P[\text{shuffle}]$ and describe how it should be used for encryption. The information s by the means of which $P[\text{shuffle}]$ is reduced to $P[\text{easy}]$ is kept as a secret trapdoor.

• Construct the cryptosystem in such a way that decryption is essentially different for the cryptanalyst and the legitimate receiver. Whilst the former must solve $P[\text{shuffle}]$, the latter may use s and solve only $P[\text{easy}]$.

Roughly at the same time when RSA was discovered [17], knapsack encryption was introduced by Merkle and Hellman [11]. It used the knapsack problem where $P[\mathsf{easy}]$ was superincreasing and shuffling was a linear operation modulo some large integer. As is well known, the knapsack cryptosystem was broken by Shamir. A variant of the knapsack system was proposed by Chor and Rivest [4] where shuffling was more elaborate since it was based on computing discrete logarithms in finite fields. Later on, building on Chor and Rivest's work, Lenstra [10] introduced the powerline system which, instead of computing logarithms, used directly the multiplicative structure of the field. For the sake of accurate paternity respect, let us stress that the construction presented in this paper uses a multiplicative version of the basic (additive) knapsack problem by combining two old, and once well-known, techniques : the multiplicative Merkle-Hellman knapsack [11] and Pohlig-Hellman's secret-key cryptosystem [15]. The new scheme therefore relates to Merkle-Hellman's cryptosystem very much the same way as the powerline system is related to the Chor-Rivest scheme. Actually, we were not aware of [10] and it is through a note by Paul Camion [3] that we understood that we had found a missing species.

The scheme presented in this article is based on the following problem :

P : given p, c and a set $\{v_i\}$, find a binary vector x such that

$$c = \prod_{i=0}^{n} v_i^{x_i} \bmod p$$

It is easy to observe that if the v_i-s are relatively prime and much smaller than p, P can be solved in polynomial time by factoring c :

$P[\mathsf{easy}]$ is an instance of P where $p > \prod_{i=0}^{n} v_i$ and $\gcd(v_i, v_j) = 1$ for $i \neq j$.

The scrambled $P[\mathsf{shuffle}]$ is obtained by extracting a secret (s-th) modular root of each v_i in $P[\mathsf{easy}]$. By raising a product of such roots to the s-th power, each v_i shrinks back to its original size and x can be found by factoring.

The following sections describe how to use P for public-key encryption.

2 The new scheme

Let p be a large public prime and denote by n the largest integer such that :

$$p > \prod_{i=0}^{n} p_i \text{ where } p_i \text{ is the } i\text{-th prime (start from } p_0 = 2)$$

The secret-key $s < p - 1$ is a random integer such that $\gcd(p-1, s) = 1$ and the public-keys are the $n+1$ roots generated *à la* Pohlig-Hellman [15] :

$$v_i = \sqrt[s]{p_i} \bmod p$$

$m = \sum\limits_{i=0}^{n} 2^i m_i \in \mathcal{M}$ is encrypted as $c = \prod\limits_{i=0}^{n} v_i^{m_i} \bmod p$ and recovered by :

$$m = \sum_{i=0}^{n} \frac{2^i}{p_i - 1} \times \Big(\gcd(p_i, c^s \bmod p) - 1 \Big)$$

Naturally, as in all knapsack-type systems, the v_is can be permuted and re-indexed for increased security.

2.1 a small example

key generation for $n = 7$ The prime $p = 9700247 > 2\times3\times5\times7\times11\times13\times17\times19$ and the secret $s = 5642069$ yield the v-list :

$$\begin{array}{ll} v_0 = \sqrt[s]{2} \bmod p = 8567078 & v_4 = \sqrt[s]{11} \bmod p = 8643477 \\ v_1 = \sqrt[s]{3} \bmod p = 5509479 & v_5 = \sqrt[s]{13} \bmod p = 6404090 \\ v_2 = \sqrt[s]{5} \bmod p = 2006538 & v_6 = \sqrt[s]{17} \bmod p = 1424105 \\ v_3 = \sqrt[s]{7} \bmod p = 4340987 & v_7 = \sqrt[s]{19} \bmod p = 7671241 \end{array}$$

encryption of $m = 202 = 11001010_2$

$$c = v_7^1 \times v_6^1 \times v_5^0 \times v_4^0 \times v_3^1 \times v_2^0 \times v_1^1 \times v_0^0 \bmod p = 7202882$$

decryption by exponentiation, we retrieve :

$$c^s \bmod p = 7202882^{5642069} \bmod 9700247 = 6783$$

whereby :

$$6783 = 19^1 \times 17^1 \times 13^0 \times 11^0 \times 7^1 \times 5^0 \times 3^1 \times 2^0 \quad \rightarrow \quad m = 11001010_2$$

information rate The information rate of our scheme (number of cleartext bits packed into each ciphertext bit) is sub-optimal since, in this example :

$$\mathcal{I} = \log(m) / \log(c) = \frac{8}{24} \simeq 33.33\% < 1$$

2.2 p as a function of n

Evaluating the growth of p and n is important for comparing and understanding the characteristics on the new scheme since message-space mainly depends on n while computational complexity is proportional to the square of p's size.

Lemma 1 *Asymptotically :*

$$p\, e^{\mathrm{li}(n)} \sim n! \log^n(n) \quad \textit{where} \quad \mathrm{li}(n) = \int_2^n \frac{dx}{\log(x)} \sim \frac{n}{\log(n)}$$

whereas interpolation for $128 \leq n \leq 418$ and $989 < \log p < 4096$ yields :

$$|1000 \log p + 144525 - n\,(8062.11 + 6.74\,n) + 4.26337\,(n/10)^3\,| < 1012$$

The following table summarises the relation between p and n for five frequent sizes of p :

size of p	n	p_n	$\mathcal{M}$	size of the v-list	$\mathcal{I}$
512 bits	74	379	75 bits	4,800 bytes	14.6 %
640 bits	88	461	89 bits	7,120 bytes	13.9 %
768 bits	103	569	104 bits	9,984 bytes	13.5 %
1,024 bits	130	739	131 bits	16,768 bytes	12.8 %
2,048 bits	232	1471	233 bits	59,648 bytes	11.4 %

Although, as explained in the next sub-section, the first three instances (512, 640 and 768) are only given for illustrative purpose.

2.3 The size of p

$\mathcal{M}$ must be sufficiently large (we recommend *at least* $n \geq 160$) to prevent birthday-search [20] through two lists of $2^{n/2}$ elements to find a couple of sets such that :

$$\prod_{i \in set[1]} v_i = \Big(\prod_{i \in set[2]} v_i \Big)^{-1} c \mod p$$

$\mathcal{M}$ and $\mathcal{I}$ can be increased by combining the following strategies :

Represent m in a non-binary base ($m = \sum_{i=0}^{n} r^i m_i, 0 \leq m_i < r$) and let

$$p > \prod_{i=0}^{n} p_i^{r-1}$$

Encryption and decryption become :

$$c = \prod_{i=0}^{n} v_i^{m_i} \bmod p \text{ and } m = \sum_{i=0}^{n} \frac{r^i}{\log\,(p_i)} \times \log \gcd(p_i^{r-1}, c^s \bmod p)$$

size of p	n	p_n	r	$\mathcal{M}$	size of the v-list	$\mathcal{I}$
1,024 bits	74	379	3	119 bits	9,600 bytes	11.6 %
2,048 bits	130	739	3	208 bits	33,536 bytes	10.2 %
2,048 bits	93	491	4	188 bits	24,064 bytes	9.2 %
2,048 bits	47	223	8	144 bits	12,288 bytes	7.0 %
2,048 bits	39	173	10	133 bits	10,240 bytes	6.5 %

Let $p < \prod_{i=0}^{n} p_i$ but restrict $\sum_{i=0}^{n} m_i = w$ so that $\forall m \in \mathcal{M},\ \prod_{i=0}^{n} p_i^{m_i} < p$.

This variant implies a non-standard coding (constant-weight messages are rather suited to random-challenge identification and less for encryption) but results in drastically smaller v-lists :

size of p	n	p_n	w	$\mathcal{M}$	**size of the v-list**	$\mathcal{I}$
512 bits	131	743	55	125 bits	8,448 bytes	24.4 %
512 bits	271	1747	47	176 bits	17,408 bytes	34.4 %
768 bits	199	1223	76	187 bits	19,200 bytes	24.3 %
768 bits	274	1777	71	222 bits	26,400 bytes	28.9 %
1,024 bits	419	2903	89	308 bits	53,760 bytes	30.1 %
1,024 bits	479	3413	87	323 bits	61,440 bytes	31.5 %

Note that it is also possible to require that $\sum_{i=0}^{n} m_i \leq w$ but this complicates coding and has a very limited effect on $\mathcal{I}$.

2.4 The arithmetic properties of p

The mulipliticative property of the Legendre symbol yields :

$$\prod_{i \in A} (-1)^{m_i} = \left(\frac{c}{p}\right) \text{ where } A = \{0 \leq i \leq n,\ \ p_i \in NQR_p\}$$

Even if the leakage of the bit :

$$b = \sum_{i \in A} m_i \mod 2$$

is not serious in itself, it may become dangerous in some specific scenari; typically, if the same m is sent to several users, relations of the form

$$b_j = \sum_{i \in set[j]} m_i \mod 2$$

can be collected and m reconstructed by linear algebra.

A trivial countermeasure would be to restrict $p_i \in QR_p$ (in this case, s can also be even)[1] but one may proceed in a more elegant way by specifying $p_0 = 2 \in NQR_p$ and *simila similibus curantur*, let

$$m_0 = \sum_{i \in A - \{0\}} m_i \mod 2$$

[1] there are exactly 54 one-byte primes, 43 nine-bit primes and 75 ten-bit primes. If one has to discard half of them, and if one wants to have a sub-minimal 160-bit message space, 50 of the primes will be eleven-bit numbers and key generation will only be possible in the lucky event where the quadratic residues have an uneven distribution and concentrate on small values.

cancel b.

Other small factors of $p-1$ produce similar phenomena. If q is such a factor, then, by raising the ciphertext to the power $(p-1)/q$, one ends up with an element of a multiplicative sub-group of order q. Since q is small, discrete logarithms can be computed in this sub-group and yield a linear equation modulo q where the message bits are the unknowns. Leakage through other factors of $p-1$ is avoided by using a safe prime *i.e.* a prime p such that $(p-1)/2$ is prime as well.

3 Some applications

3.1 Processing encrypted data

A major weakness of software encryption is that while being processed, data are in a vulnerable state. For being modified, information must be deciphered and re-encrypted again. Unfortunately, while in clear, secrets are exposed to a wide gamut of threats ranging from scanning by hostile TSR-programs to interception in residual electromagnetic radiation.

The new cryptosystem seems interesting for processing encrypted data as it allows to modify (only) m_k by multiplying (or dividing) c by v_k. If $m_k = 1$, an additional multiplication by v_k is likely to have no effect on the cleartext[2] but if $m_k = 0$, modular division (by v_k) will destroy the whole plaintext.

3.2 Incremental encryption

Similarly, the sender can pre-encrypt a chunk of m and complete c later. This feature can be used in group-encryption protocols where each participant adds an encrypted chunk to a common ciphertext without gaining knowledge about the chucks encrypted by his peers (again, each chunk should be sufficiently big to avoid exhaustive search and properly protected against modular division).

When protection against active attacks is needed (that is, when the peers are malicious active adversaries), this feature can be inhibited by using a part of m as a (sufficiently big) CRC or by pre-encrypting m with a conventional block-cipher keyed with some public constant.

3.3 Batch encryption

Surprisingly, encrypting a pair of random message-blocks (here $m[1]$ and $m[2]$) requires only 75% of the multiplications needed for two sequential encryptions $(i = 1, 2)$:

$$c[i] = \mathsf{encrypt}(m[i] \oplus m[1] \wedge m[2]) \ \times \mathsf{encrypt}(m[1] \wedge m[2]) \bmod p$$

Although this strategy can be generalised to more than two blocks by building an intersection tree, accurate evaluation indicates that bookkeeping quickly costs the gain.

[2] the probability that $p_k \prod_{i=0}^{n} p_i^{m_i} < p$ is very close to one if m is uniformly distributed.

4 Implementation

In order to fit into a 68HC05-based ST16CF54 smart-card (4,096 EEPROM bytes, 16,384 ROM bytes and 352 RAM bytes), key storage was replaced by a command that re-computes the v-list upon request (re-computation and transmission take 310 ms per v_i but have to be done only once after reset). The p-list is compressed into a string of 48 bytes (in our implementation, $n = 74$) which k-th bit equals one if and only if k is prime. p_i is extracted by scanning this string until i ones were read (p_i is then the value of the scan-counter). To speed-up decryption (215 ms plus 33 ms for DES pre-encryption), our 824-byte program uses a composite p (four 256-bit factors) and sub-contracts all base-conversion operations ($r = 3$) to the smart-card reader. Benchmarks were done with a 5 MHz oscillator and ISO 7816-3 T=0 transmission at 115,200 bauds.

As strange as it may appear, the PC encrypts RSA-compatible ciphertexts without using a public exponent. Publishing $e = 1/s \bmod \phi(p)$ will make the computation of the v-list public but result in a standard RSA with a particular message format.

Although we see no immediate objection to restrict s to 160 bits, we recommend to avoid doing so before a reasonable scrutiny period (in particular, using a short s with a composite p seems related to [24, 23]) and enforce, in general, the following recommendations :

• As for any block cipher, too short messages ($\leq$ 64 bits) should not be encrypted, unless concatenated to an appropriate randomiser [6].

• As for RSA and DSA [9], correct implementation must hide the correlation between processing time and the weights of m and s.

• To avoid oracle attacks [1], we recommend to reject all decrypted messages that, when re-encrypted *by the receiver* do not re-yield c.

• Since the p-list is not necessary for encryption, we recommend to keep it secret in practice but assume its knowledge as a weakened target for the sake of academic research.

Unlike RSA, our scheme is not patented; hardware and software implementing the cryptosystem can therefore be freely used and disseminated.

5 Challenge

It is a tradition in the cryptographic community to offer cash rewards for successful cryptanalysis. More than a simple motivation means, such rewards also express the designers' confidence in their own schemes. As an incentive to the analysis of the new scheme, we therefore offer (as a souvenir from Eurocrypt'97) DM 1024 to whoever will decrypt :

c = 9D581F9E996C5D0878DC92BF5D5A8D2177B8B853E6697007
47D2C1411FAC6346045C76596193DE57A3996F04395E7BD44780
157CE4497E506DA61F09B73BAF3286272AC1625A5D989749BD38
46B634819BD26DF278CF6CD9157B891C629D3ECB49CB6E18D57E
4D9D4B70DA14738E1654F7466B48A0FCF96E0A7CBEF7A7A05DDA$_{16}$

$$
\begin{aligned}
p = \texttt{ } & \texttt{EB17673456CF46F2F819B1FB5B15D330FCF1BB063E6C5DBB} \\
& \texttt{A2A675D1639F0AF897C6CF04B3DEE33EBA5795C4A2E7EEF7CD28} \\
& \texttt{5721B97F184159987F91DDC9C8270E5D36B2562F23B3881DD795} \\
& \texttt{FB53634679944F3F11027B1D90BB8D3767151069626420E64E02} \\
& \texttt{029BE0FA5ECEFC6987C72C10451CC033FFD77A78E8B8B2A60623}_{16}
\end{aligned}
$$

where $r = 4$, $n = 74$ and the coding convention is $\texttt{space} = 0$, $\texttt{a} = 1$, $\texttt{b} = 2, \cdots, \texttt{z} = 26$. The challenger should be the first to decrypt at least 50% of c (the v-list is available by email) and publish the cryptanalysis method which must be different than computing the discrete logarithm of one of the v_i-s but the authors are ready to carefully evaluate *ad valorem* any feedback they get.

6 Further research

Since a first (informal) presentation of the scheme, several researchers began to investigate its different aspects and compare its features to RSA [5, 12, 2].

Elliptic curving the scheme is still an open problem (since elliptic curves are Abelian groups and not Euclidean domains, gcds can not be computed). Provable security, strategies for reducing the size of the public-key or signing with the scheme are also important for increasing the *practical* usefulness of the new cryptosystem.

A general knapsack taxonomy also seems in order. The idea of multiplicative knapsack is roughly 20 years old and was first proposed in the open literature by Merkle and Hellman [11] in their original paper. As, observed by Desmedt in his 1986 survey [7], encryption in the multiplicative Merkle-Hellman knapsack is actually additive. It is in fact the decryption which is multiplicative. The scheme presented here is in this respect thoroughly multiplicative. It should also be noted that Merkle-Hellman's knapsack was (partially) cryptanalyzed in by Odlyzko [13] but all our attempts to extend this attack to the new scheme failed.

As a final conclusion, although our scheme seems practical and simple, it can hardly compete with RSA on concrete commercial platforms as its public keys are typically eighty times bigger than RSA ones. Nevertheless, the new concept appears to be a promising starting-point for improvements and further research.

7 Acknowledgements

The authors thank Yvo Desmedt, Philippe Hoogvorst, David Kravitz and Ronald Rivest and Eurocrypt's referees for helpful comments and discussions.

References

1. R. Anderson, *Robustness principles for public-key protocols*, LNCS, Advances in Cryptology, Proceedings of Crypto'95, Springer-Verlag, pp. 236–247, 1995.
2. R. Anderson & S. Vaudenay, *Minding your p's and q's*, LNCS, Advances in Cryptology, Proceedings of Asiacrypt'96, Springer-Velrag, pp. 26–35, 1996.

3. P. Camion, *An example of implementation in a Galois field and more on the Naccache-Stern public-key cryptosystem*, manuscript, October 27–29, 1995.
4. B. Chor & R. Rivest, *A knapsack-type public key cryptosystem based on arithmetic on finite fields*, IEEE Transactions on Information Theory, vol. IT 34, 1988, pp. 901–909.
5. T. Cusick, *A comparison of RSA and the Naccache-Stern public-key cryptosystem*, manuscript, October 31, 1995.
6. D. Denning (Robling), *Cryptography and data security*, Addison-Wesley Publishing Company, p. 148, 1983.
7. Y. Desmedt, *What happened with knapsack cryptographic schemes*, Performance limits in communication - theory and practice, NATO ASI series E : Applied sciences, vol. 142, Kluwer Academic Publishers, pp. 113-134, 1988.
8. W. Diffie & M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory, vol. IT 22 n° 6, pp. 644–654, 1976.
9. P. Kocher, *Timing attacks in implementations of Diffie-Hellman, RSA, DSS and other systems*, LNCS, Advances in Cryptology, Proceedings of Crypto'96, Springer-Verlag, pp. 104–113, 1996.
10. H. Lenstra, *On the Chor-Rivest knapsack cryptosystem*, Journal of Cryptology, vol. 3, pp. 149–155, 1991.
11. R. Merkle & M. Hellman, *Hiding information and signatures in trapdoor knapsacks*, IEEE Transactions on Information Theory, vol. IT 24 n° 5, pp. 525–530, 1978.
12. M. Naor, *A proposal for a new public-key by Naccache and Stern*, presented at the Weizmann Institute Theory of Computation Seminar, November 19, 1995.
13. A. Odlyzko, *Cryptanalytic attacks on the multiplicative knapsack cryptosystem and on Shamir's fast signature scheme*, IEEE Transactions on Information Theory, vol. IT 30, pp. 594–601, 1984.
14. H. Petersen, *On the cardinality of bounded subset products* , Technical report TR-95-16-E, University of Technology Chemnitz-Zwickau, 1995.
15. S. Pohlig & M. Hellman, *An improved algorithm for computing logarithms over GF(q) and its cryptographic significance*, IEEE Transactions on Information Theory, vol. 24, pp. 106–110, 1978.
16. D. Pointcheval, *A new identification scheme based on the perceptrons problem*, LNCS, Advances in Cryptology, Proceedings of Eurocrypt'94, Springer-Verlag, pp. 318–328, 1995.
17. R. Rivest, A. Shamir & L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, CACM, vol. 21, n°. 2, pp. 120-126, 1978.
18. A. Salomaa, *Public-key cryptography*, EATCS Monographs on theoretical computer science, vol. 23, Springer-Verlag, page 66, 1990.
19. A. Shamir, *An efficient identification scheme based on permuted kernels*, LNCS, Advances in Cryptology, Proceedings of Crypto'89, Springer-Verlag, pp. 606–609.
20. G. Simmons, *Contemporary cryptology : The science of information integrity*, IEEE Press, pp. 257–258, 1992.
21. J. Stern, *A new identification scheme based on syndrome decoding*, LNCS, Advances in Cryptology, Proceedings of Crypto'93, Springer-Verlag, pp. 13–21, 1994.

22. J. Stern, *Designing identification schemes with keys of short size*, LNCS, Advances in Cryptology, Proceedings of Crypto'94, Springer-Verlag, pp. 164–173, 1994.
23. P. van Oorschot & M. Wiener, *On Diffie-Hellman key agreement with short exponents*, LNCS, Advances in Cryptology, Proceedings of Eurocrypt'96, Springer-Verlag, pp. 332–343, 1996.
24. M. Wiener, *Cryptanalysis of short RSA secret exponents*, IEEE Transactions on Information Theory, vol. 36, n^o. 3, pp. 553–558, 1990.