Designing Identification Schemes with Keys of Short Size ⋆.

Jacques Stern

Laboratoire d'Informatique, École Normale Supérieure

Abstract. In the last few years, there have been several attempts to build identification protocols that do not rely on arithmetical operations with large numbers but only use simple operations (see [10, 8]). One was presented at the CRYPTO 89 rump session ([8]) and depends on the so-called Permuted Kernel problem (PKP). Another appeared in the CRYPTO 93 proceedings and is based on the syndrome decoding problem (SD) form the theory of error correcting codes ([11]). In this paper, we introduce a new scheme of the same family with the distinctive character that both the secret key and the public identification key can be taken to be of short length. By short, we basically mean the usual size of conventional symmetric cryptosystems. As is known, the possibility of using short keys has been a challenge in public key cryptography and has practical applications. Our scheme relies on a combinatorial problem which we call *Constrained Linear Equations* (CLE in short) and which consists of solving a set of linear equations modulo some small prime q, the unknowns being subject to belong to a specific subset of the integers mod q. Thus, we enlarge the set of tools that can be used in cryptography.

1 The Underlying Problem

Since the appearance of public-key cryptography, basically all practical schemes have been based on hard problems from number theory. This has remained true with zero-knowledge proofs, introduced in 1985, in a paper by Goldwasser, Micali and Rackoff ([6]) and whose practical significance was soon demonstrated in the work of Fiat and Shamir ([4]). In 1989, there were two attempts to build identification protocols that only use simple operations (see [10, 8]). One relied on the intractability of some coding problems, the other on the Permuted Kernel problem (PKP). The first of the schemes was not really practical but has been followed by a truly practical proposal based on the so-called Syndrome Decoding problem (SD). The purpose of the present paper is twofold:

– First, to introduce a new scheme based on a combinatorial problem which we call *Constrained Linear Equations* (CLE in short) and which consists of solving a set of linear equations modulo some small prime q, the unknowns being subject to belong to a specific subset of the integers mod q.

⋆ PATENT CAUTION: This document may reveal patentable subject matter

– Second, to demonstrate in this setting, the possibility to have an identification scheme where both the secret key and the public identification key can be taken to be of short length. By short, we basically mean the usual size of conventional symmetric cryptosystems, i.e. 64 or 80 bits.

We briefly comment on the second point. Besides having been a long time challenge in public key cryptography, the question of short keys may be of practical importance. As is known, identification schemes avoiding large integers such as PKP or SD, are not identity based. This means that public keys should be related to the user's identity by a signature of some authority or by a directory, which the verifier has access to. If the second option is taken, then the key length becomes an important issue. It is even more important for the prover, a smart card in many practical applications: short keys may save space in the physically protected area of the card where they are stored and thus may allow the use of relatively low cost cards.

We now turn to our basic problem:

Constrained Linear Equations (CLE)
instance: A (small) prime number q, a system S of r homogeneous linear equations with k unknowns and whose coefficients are integers mod q, a subset X of the integers mod q.
Question Is there a solution of S consisting of k elements of the given set X?

It is easily seen that the problem is $\mathcal{N}P$-complete. Our further assumption is that it is intractable in the following sense:

Intractability of CLE Assumption: No probabilistic polynomial time algorithm can take as its input the values of q, S, X and output, with non negligible probability, a solution of S consisting of k elements of the given set X.
As usual, non negligible stands for bounded from below by the inverse of some power of the size of the input.

From the practical point of view, we mention, as a minimal choice, the case where $q = 257$, $k = 40$, $r = 20$, $|X| = 16$. We do not really advocate these figures for highly secure applications but we use them as a convenient benchmark in order to establish comparisons with the minimal sizes provided for PKP or SD. The minimal size suggested for the SD identification scheme has been carefully analyzed in [3], where it is shown that the workload of the best possible known attacks is about 2^{68}. The minimal size of the parameters in the original PKP proposal (see [8]) has been extensively discussed in [1, 7]. Attacks based on intelligent gaussian elimination and a space-time trade off yield a workload of 2^{52}. Similar attacks can be carried against CLE and it can be seen that the figures chosen above yield a similar 2^{52} workload. Thus, the comparisons in terms of key size will be significant. Whether or not this is enough for applications is open for discussion. We feel that, for secure applications, it is safer to recommend the following parameters for CLE: $q = 257$, $k = 48$, $r = 24$, $|X| = 16$.

2 Key Generation

The key generation algorithm is based on a trick which combines a linear operation and a highly non linear one. We feel that this trick might find further applications in other areas of cryptography. The prime number q is such that $q-1$ is the product of two almost equal integers. Thus, 251 is $15*16+1$ and 257 is 16^2+1. We write $q-1=c*d$. We next consider the multiplicative group of non-zero integers mod q. This is a cyclic group and we can easily build a subgroup G of order c. Picking one element in each class mod G, we get a set X, consisting of d elements such that any integer between 1 and $q-1$ can be uniquely written as the product mod q of an element of G and an element of X. We let $g(u)$ the element of G appearing in the unique decomposition of u and, similarly, we let $k(u)$ be the corresponding element of X. If U is a vector whose coordinates are non-zero integers mod q, we let $g(U)$ be the vector obtained by applying g coordinatewise. $k(U)$ is defined accordingly.

Besides a fixed prime number q, a subgroup G and a fixed subset X as above, the proposed scheme uses a fixed $(n \times m)$-matrix M whose coefficients are randomly chosen integers mod q. This matrix is common to all users and is originally built randomly. Each user receives a secret key S which is a vector with m coordinates, each a member of X. The public identification is computed as

$$P = g(M(S))$$

Note that there is a (slight) chance that the computation of $g(M(S))$ cannot be carried through if $M(S))$ has some zero coordinate.A heuristic analysis shows that this happens with probability close to $1-(1-\frac{1}{q})^n$. With the figures of the numerical example provided above, this is 0.07 and therefore, after a few trials, one can reach the desired value of P.

Clearly, recovering the secret key S from the public data P amounts to solving the equation

$$P \otimes T = M.S$$

where $\otimes$ denotes coordinatewise multiplication mod q and where S, T are two unknown vectors, having respectively m and n coordinates, subject to the condition that these coordinates are members of X . Thus, one has to solve an instance of the CLE problem with n equations and $m+n$ unknowns.

If we turn to our minimal size numerical example, we see that one can take $q = 257$ and $n = m = 20$. Furthermore, X has 16 elements as well as G. Thus both S and P can be coded on 80 bits. Without apparent consequence on the intractability of the combinatorial problem to solve, S can be generated deterministically from a (say) 64-bit seed. It is also possible to fix the first four coordinates of the public key P or (better) to derive them from the public identity of the user. The resulting public key is stored on 64 bits. The key generation uses, as above, the idea of multiple trials. The expected number of trials until an acceptable key is found is about $7*10^4$, which is still reasonable. A few more bits could even be saved by analogous tricks. We feel that these manipulations do not affect the security of the scheme that we will present, but this opinion should be further investigated. If we turn to alternative numerical example mentioned above, we find the following figures: $q = 257$, $n = m = 24$, $|X| = |G| = 16$. Using the tricks just described, this is still compatible with secret and public keys of 80 bits.

We will now describe, in the style of [8, 11], two interactive identification protocols by which a prover demonstrates possession of the secret key S corresponding to the public key P.

3 A Three Pass Identification Protocol

As is the case for the PKP and SD schemes, we will need some cryptographic hash function H. This hash function should be collision free as will be discussed further. For practical implementations, a standard hash function such as Rivest's MD5 ([9]) can be used. The protocol includes several rounds, each of these being performed as follows:

1. The prover picks at random two vectors U, V having respectively m and n coordinates, each an integer mod q. He also chooses two random permutations σ and τ. σ operates on the integers $\{1 \cdots m\}$ and τ on the integers $\{1 \cdots n\}$. Then he sends commitments h_1, h_2, h_3 respectively computed as
$$h_1 = H(\sigma, \tau, M.U + P \otimes V)$$
$$h_2 = H(U.\sigma, V.\tau)$$
$$h_3 = H((U + S).\sigma, (V - T).\tau)$$
In the above, H denotes the cryptographic hash function, $+$, $-$ and $\otimes$ denote coordinatewise operations mod q and $U.\sigma$ stands for the action of σ on U, that is to say the vector $U_{\sigma(i)}$, $1 \leq i \leq m$. Also, T is the vector with n coordinates defined by $T = k(M(S))$, with the notations of section 2.
2. The verifier sends a random element b of $\{0, 1, 2\}$.
3. If b is 0, the prover reveals σ, τ U and V. If b is 1, the prover reveals σ, τ and the two vectors $U' = (U + S)$ and $V' = (V - T)$. Finally, if b equals 2, the prover discloses vectors $U.\sigma$, $V.\tau$ together with vectors $U" = (U + S).\sigma$ and $V" = (V - T).\tau$.
4. If b equals 0, the verifier checks that commitments h_1 and h_2 have been computed honestly. This is possible since, using the values of σ, τ, U and V disclosed at step 2, he can compute the respective values of $M.U + P \otimes V$, $U.\sigma$, $V.\tau$. From these values he checks that $h_1 = H(\sigma, \tau, M.U + P \otimes V)$ and that $h_2 = H(U.\sigma, V.\tau)$.
If b equals 1, the verifier checks that commitments h_1 and h_3, were correct: note that σ, τ are known from step 3 and that
$$M.U + P \otimes V = M.(U + S) + P \otimes (V - T) - M.S + P \otimes T = M.U' + P \otimes V'$$
This allows the verifier to check the equality $h_1 = H(\sigma, \tau, M.U' + P \otimes V')$. He can also check that $h_3 = H(U'.\sigma, V'.\tau)$.

Now, if b is 2, the verifier checks commitments $h_2 = H(U.\sigma, V.\tau)$ and $h_3 = H(U", V")$. Furthermore, the verifier computes the two vectors $U'' - U.\sigma$ and $V.\tau - V''$ and verifies that all of their coordinates are members of X.

The number r of consecutive rounds depends on the required level of security and will be discussed further on.

4 A Five Pass Identification Protocol

We new describe an alternative protocol allowing identification. Again, this protocol includes several rounds, each of these being performed as follows:

1. The prover picks at random two vectors U, V having respectively m and n coordinates, each an integer mod q. He also chooses two random permutations σ and τ. σ operates on the integers $\{1 \cdots m\}$ and τ on the integers $\{1 \cdots n\}$. Then he sends commitments h_1, h_2 respectively computed as
$$h_1 = H(\sigma, \tau, M.U + P \otimes V)$$
$$h_2 = H(S.\sigma, T.\tau, U.\sigma, V.\tau)$$
In the above, all notations are as in the previous section.
2. The verifier sends a random element a of between 0 and $q-1$.
3. The prover computes the pair $Y = (aS + U).\sigma$, $Z = (aT - V).\tau$ and sends back these two vectors to the verifier.
4. The verifier sends a random bit b of $b = 0$ or 1.
5. If b is 0, the prover reveals σ and τ. If b is 1, the prover discloses vectors $S.\sigma$ and $T.\tau$
6. If b equals 0, the verifier checks commitments h_1. This is possible since, using the values of σ and τ disclosed at step 2, he can compute successively the respective values of $Y' = Y.\sigma^{-1}$ and $Z' = Z.\tau^{-1}$, and then $M(Y' - P \otimes Z')$. Provided the answer is correct this last vector equals $M.U + P \otimes V$, From these values he checks that $h_1 = H(\sigma, \tau, M.U + P \otimes V)$.
If b equals 1, the verifier checks that commitment h_2 was correct: note that if the correct values of $S.\sigma$ and $T.\tau$ have been received, the verifier can compute vectors $Y - aS.\sigma$ and $aT.\tau - Z$. These are respectively equal to $U.\sigma$ and $V.\tau$, so that one should have $h_2 = H(S.\sigma, T.\tau, Y - aS.\sigma, aT.\tau - Z)$. Having checked this equality, the verifier also tests that the vectors received at step 5 are such that all of their coordinates are members of X.

5 Security of the Scheme

It is apparent that the security of the scheme relies on the difficulty of solving the equation
$$P \otimes T = M.S$$

where P is the public key of a specific user and S, T are two unknown vectors, having respectively m and n coordinates, subject to the condition that these coordinates are members of X .

In order to perform the first protocol without knowing the secret key, various strategies can be used.

- Having only U, V, σ and τ ready for the verifier's query and replacing the unknown S, T by arbitrary vectors with coordinates in X. In this case, the false prover hopes that b is 0 or 2 and the probability of success is 2/3 for a single round and $(2/3)^r$ in general, where r is the number of rounds. A similar strategy can be defined with $U+S$, $V-T$ in place of U, V.
- Having simultaneously U, V, $U+S'$ and $V-T'$ ready where S', T' is a regular solution of
$$P \otimes T = M.S$$
(i.e. without the constraint about X). This yields the same probability of success.

It is fairly clear that shifting beetween one strategy to another has also the same probability of success. Similar strategies can be designed for the second protocol with probability of success $\frac{q+1}{2q}$ and $\left(\frac{q+1}{2q}\right)^r$ if the protocol is repeated r times. In the reverse direction, we have:

Theorem 1. *Assume that some probabilistic polynomial-time adversary is accepted with probability $\geq (2/3)^r+\epsilon$ after playing a constant number r of rounds of the first protocol with a fair verifier , then there exists a polynomial-time probabilistic machine which extracts a secret pair S, T from the public data or output collisons for the hash function, with overwhelming probability.*

remark There is an analogous result for the second protocol with $(2/3)^r$ replaced by $\left(\frac{q+1}{2q}\right)^r$.
proof: Consider the tree $T(\omega)$ of all 3^r executions corresponding to all possible questions of the verifier when the adversary has a fixed random tape ω. Let
$$\alpha = Pr(T(\omega) \qquad \text{has a vertex with 3 sons} \qquad)$$
If α is $<\epsilon$, then, it is easily seen that the probability of succes of the adversary is bounded by $(2/3)^r+\epsilon$: $(2/3)^r$ comes from the case where $T(\omega)$ has no vertex with 3 sons and ϵ from the other case. Thus α is at least ϵ and by resetting the adversary $1/\epsilon$ times, one finds, with constant probability an execution tree with a vertex having 3 sons. Repeating again, the probability can be made very close to one. Now a vertex with 3 sons corresponds to a situation where 3 commitments h_1, h_2, h_3 have been made and where the adversary can provide answers to the 3 possible queries of the verifier. Consider the answer σ, τ, U, V to the question $b=0$ and the answer σ', τ', U', V' to the question $b=1$. Since
$$H(\sigma, \tau, M.U + P \otimes V) = h_1 = H(\sigma', \tau', M.U' + P \otimes V')$$

we conclude that either a collision for the hash function H has been found or else $\sigma = \sigma'$, $\tau = \tau'$ and $M.U + P \otimes V = M.U' + P \otimes V'$. Similar arguments show that, unless an H-collision has been found, the answer to $b = 2$ consists of $U.\sigma$, $V.\tau$, $U'.\sigma$, $V'.\tau$. We note that, since the last answer is accepted, both $(U' - U).\sigma$ and $(V - V').\tau$ have all coordinates in X. Also $M.(U' - U) = P \otimes (V - V')$, as observed above. It follows that the underlying system of constrained linear equations has been solved.

Following the techniques in [5], it is possible to prove a more foundational result, which shows that repetition of either protol is a *proof of knowledge* of a solution of the constrained system

$$P \otimes T = M.S$$

We state such a result for our first protocol. We let N denote the size of the public data.

Theorem 2. *Assume that some probabilistic polynomial-time adversary is accepted with non negligible probability after playing with a fair verifier a number of rounds of the first protocol that is $\Theta(\log N)$, then there exists a polynomial-time probabilistic machine which extracts a secret pair S, T from the public data or outputs collisons for the hash function, with overwhelming probability.*

Before we turn to zero-knowledge, let us observe that, at step 3 of the first protocol, the prover eventually discloses the image of the secret pair S, T, under two random permutations σ and τ. A similar remark applies to the other protocol. Thus, the exact repartition of the values of the unknowns in S and separately in T have to be considered as public data. This information makes the computation of the solutions of

$$P \otimes T = M.S$$

a bit easier. We have taken this into account when analyzing the security of the CLE problem. Still, it is advisable to avoid irregular distributions (e.g. where an element of S appears many times).

It can be proved formally that both schemes are zero-knowledge. We will only give a brief hint for the first protocol. As we observed above, anyone can be ready to answer two queries among the three possible ones at each round. Hence, by using the standard idea of resettable simulation (see [6]), one can devise a polynomial-time simulation algorithm that mimics the fair communication between the prover and the verifier in expected time $O(2/3.r)$. Some remarks are in order here:

1. As was just observed, the exact repartition of the values of the unknowns in S and T are basically public data. This does not contradict zero-knowledge as they leak equally from the actual executions and the simulated ones.
2. Hash values make the simulation a bit harder: a convenient setting is the so-called *random oracle model* (see [2]). Alternatively, one has to assume specific statistical independance properties for the hash function.

6 Performances of the Scheme.

The performances of our scheme are very comparable to those of [8, 11] and we will restrict ourselves to various remarks.

1. As for previous schemes of the same family, the memory needed to implement the scheme is not large: especially, it is not necessary to store all of M. One can only store words corresponding to some chosen locations and extend these by a fixed software random number generator.
2. The operations to perform are very simple and well suited to the environment of 8-bit microprocessors.
3. The communication complexity of the protocol is quite acceptable: if we assume that hash values are 128 bits long, we obtain an average number of bits per round which is close to 840 bits for the first protocol and 725 for the second. This is for the minimal suggested size of parameters. For the alternate choice, these figures go up to 940 and 824. There is a trick that can save one hash value, at least for the first protocol. It consists of replacing h_1, h_2, h_3 by $H(h_1, h_2, h_3)$ and providing the missing hash value at step 3 (for example transmitting h_3 if $b = 0$). This yields similar communication complexities for both schemes.
4. In order to achieve a level of security of 10^{-6}, the first protocol has to be repeated 35 times and the second one only 20 times. Whether or not this is a serious drawback should be discussed with practical implementations in mind. We simply note that the number of interactions is almost the same in both case, because the second protocol needs more passes.
5. As is the case for PKP and SD, our scheme is not identity based. This means that public keys have to be certified by the issuing authority or that the verifier needs to access a directory. As emphasized in the introduction, the distinctive character of the scheme, namely the short key length, is a definite advantage in the latter case.

7 An Additive Variant

Before concluding the paper, we briefly mention an alternative approach for key generation: let X and Y be two subsets of the set of integers mod q, such that any integer can be written (non necessarily in a unique way) as the sum mod q of an element of X and an element of Y (it is not difficult to construct such subsets). From a random vector S xith m coordinates, all in X, one can compute $M.S$ as $T + P$, where T, P are vectors with n coordinates respectively in X and Y. One can take P as a public key and keep S, T secret. The resulting CLE problem is written

$$M.S = T + P$$

where S, T are unknown vectors with coordinates in X. Protocols to prove knowledge of the secret data are simple variants of those described above. We do not know whether this alternative key generation method is weaker than the original one. We suspect it might be the case if X and Y are chosen in a simple way (e.g. by specifying that elements of X are those with prescribed bits equal to zero).

8 Conclusion

We have defined a new practical identification scheme based on a combinatorial problem which we call CLE (Constrained Linear Equations). This scheme allows the use of keys of short length (64 or 80 bits). We have proposed two protocols using CLE: both only use very simple operations and thus widen the range of techniques that can be applied in cryptography. We welcome attacks from readers and, as is customary when introducing a new cryptographic tool, we suggest that the scheme should not be adopted prematurely for actual use.

Acknowledgements

I wish to thank A. Shamir for his comments on an earlier presentation of this work and for suggestions which led to the alternative approach mentioned in section 7. I also want to thank my student J.B. Fischer for help in the evaluation of the security of CLE.

References

1. T. Baritaud, M. Campana, P. Chauvaud and H. Gilbert: On the security of the permuted kernel identification scheme. In: Proceedings of Crypto 92. Lecture Notes in Computer Science 740. Berlin: Springer 1993, pp. 305-311.
2. M. Bellare and P. Rogaway: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, 1993, pp. 62-73.
3. F. Chabaud: On the security of some cryptosystems based on error-correcting codes. In: Proceedings of Eurocrypt 94. Lecture Notes in Computer Science, to appear.
4. A. Fiat and A. Shamir: How to prove yourself: Practical solutions to identification and signature problems. In: Proceedings of Crypto 86. Lecture Notes in Computer Science 263. Berlin: Springer 1987, pp. 181-187.
5. U. Feige, A. Fiat and A. Shamir: Zero-knowledge proofs of identity. In: Proc. 19th ACM Symp. Theory of Computing, 1987, pp. 210-217 and *J. Cryptology* **1**, 77-95 (1988).
6. S. Goldwasser, S. Micali and C. Rackoff: The knowledge complexity of interactive proof systems. In: Proc. 17th ACM Symp. Theory of Computing, 1995, pp.291-304.
7. J. Patarin and P. Chauvaud: Improved algorithms for the permuted kernel prolem. In: Proceedings of Crypto 93, Lecture Notes in Computer Science 773. Berlin: Springer 1994, pp. 391-402.
8. A. Shamir: An efficient identification scheme based on permuted kernels. In: Proceedings of Crypto 89. Lecture Notes in Computer Science 435. Berlin: Springer 1990, pp. 606-609.

9. R. L. Rivest: The MD5 Message Digest Algorithm. In: Proceedings of Crypto 90. Lecture Notes in Computer Science 537. Berlin: Springer 1991, pp. 303-311.
10. J. Stern: An alternative to the Fiat-Shamir protocol. In: Proceedings of Eurocrypt 89. Lecture Notes in Computer Science 434. Berlin: Springer 1990, pp. 173-180.
11. J. Stern: A new identification scheme based on syndrome decoding. In: Proceedings of Crypto 93. Lecture Notes in Computer Science 773. Berlin: Springer 1994 , pp. 13-21.