Efficient One-Time Signatures from Quasi-Cyclic Codes

Edoardo Persichetti
Department of Mathematical Sciences, Florida Atlantic University
epersichetti@fau.edu

ABSTRACT

The design of a practical code-based signature scheme is an open problem in post-quantum cryptography. In this paper, we propose a simple and efficient scheme which is one-time secure against chosen-message attackers. The scheme stems from a framework introduced by Lyubashevsky and uses techniques from Pointcheval and Stern to achieve provable security. The use of quasi-cyclic codes guarantees compact keys and signature sizes and efficient arithmetic, making the scheme appealing in lightweight applications.

ACM Reference Format:
Edoardo Persichetti. 2018. **Efficient One-Time Signatures, from Quasi-Cyclic Codes. In *11th International Conference On Security Of Information and Networks (SIN '18), September 10–12, 2018, Cardiff, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3264437.3274832**

1 INTRODUCTION

Digital signatures are a very important cryptographic protocol in the modern world. Many schemes based on coding theory have been proposed over the years, following either a "direct" hash-and-sign approach like CFS [7] and KKS [13], or the Fiat-Shamir transform [9] to convert an identification scheme into a signature scheme. The latter are usually built via a multi-pass protocol [4, 25], relying on the seminal work of Stern [23, 24]. Unfortunately, many proposals have been broken, and all those that are still considered secure suffer from one or more flaws, be that a huge public key, a large signature or a slow signing algorithm, which make them highly inefficient in practice. This is particularly evident in the identification schemes, where it is usually necessary to repeat the protocol many times in order to guarantee correctness or security.

In this paper, we propose a code-based signature scheme that follows a different approach, inspired by the work of Lyubashevsky [14, 15]. Such a proposal had been attempted before (see [19]) without success, the main issue being the choice of the setting (random binary codes) which proved to be too restrictive. Our scheme is based on random binary quasi-cyclic codes, and includes a proof of security that guarantees one-time existential unforgeability against chosen-message attacks, i.e. 1-EUF-CMA. While one-time signatures are not used directly in most applications, they are still relevant since they can be embedded in a Merkle tree structure to obtain a full-fledged signature scheme, which allows to sign up to a predetermined number of times. Our scheme compares very well to other one-time code-based proposals, obtaining what are, to date, the smallest sizes for both signature and public data.

SIN '18, September 10–12, 2018, Cardiff, United Kingdom

ACM ISBN 978-1-4503-6608-3/18/09…$15.00
https://doi.org/10.1145/3264437.3274832

2 PRELIMINARIES

A digital signature scheme is defined by a triple of algorithms (KeyGen, Sign, Ver). The *key generation algorithm* KeyGen takes as input a security parameter λ and outputs a signing key sgk and a verification key vk. The private *signing algorithm* Sign receives as input a signing key sgk and a message m and returns a signature σ. Finally, the public *verification algorithm* Ver uses a verification key vk to verify a signature σ that is transmitted together with the message m: it outputs 1, if the signature is valid, or 0 otherwise.

The standard notion of security for digital signatures schemes is Existential Unforgeability under Chosen-Message Attacks (EUF-CMA), as described, for example, in [12]. In this scenario, the goal of an attacker is to be able to produce a valid message/signature pair, and the attack model allows the attacker to obtain a certain, predetermined, number of signatures on arbitrarily chosen messages (signing queries). In particular, if the attacker is only allowed to obtain a single signature, we talk about 1-EUF-CMA security.

An identification scheme is a protocol that allows a party $\mathcal{P}$, the Prover, to prove to another party $\mathcal{V}$, the Verifier, that he possesses some secret information x, usually called *witness*, without revealing to the verifier what that secret information is. Fiat and Shamir in [9] showed how to obtain a full-fledged signature scheme from an identification scheme. With this paradigm, the signer simply runs the identification protocol, where, for the purpose of generating the challenge, the verifier is replaced by a random oracle $\mathcal{H}$ (usually a cryptographic hash function). The signature is accepted according to the validity of the response in the identification scheme.

Table 1: The Fiat-Shamir Signature Scheme.

Setup	Select an identification scheme $\mathcal{I}$.
Sign	On input the private key of $\mathcal{I}$ and a message m, commit Y, set $c = \mathcal{H}(Y, m)$, compute a response z and return the signature $\sigma = (Y, z)$.
Ver	On input the public key of $\mathcal{I}$, a message m and a signature σ, set $c = \mathcal{H}(Y, m)$ then output 1 if z is accepted in $\mathcal{I}$, else return 0.

Note that several signature schemes, including [15] and this work, use a slightly modified version of the above paradigm, where the signature is (c, z) instead of (Y, z). The verifier can then calculate Y from z and the public key, and check the equality between c and the hash digest obtained using this newly-generated Y and m.

2.1 Quasi-Cyclic Codes

Cyclic codes, i.e. codes that can be generated by cyclic shift of a generator polynomial, have been shown to be insecure in the context of cryptography, as they introduce too much recognizable structure. A subfamily, known as *quasi-cyclic* codes, has been then proposed with some success, mostly in the context of encryption.

Definition 1. *Let C be an $[n, k]$ linear code over $\mathbb{F}_q$. We call C* Quasi-Cyclic *if there exists n_0 such that, for any codeword a all the right shifts of a by n_0 positions are also codewords.*

When $n = n_0 p$, it is possible to have both generator and parity-check matrices in a special form, composed of n_0 circulant $p \times p$ blocks. The algebra of quasi-cyclic codes can be connected to that of the polynomial ring $\mathbb{F}_q[X]/(X^p - 1)$, where each codeword is a length-n_0 vector of elements of the ring.

For the remainder of the paper, we consider only binary codes, thus we set $\mathcal{R} = \mathbb{F}_2[X]/(X^p - 1)$, and we restrict our attention to the case $n_0 = 2$. We then have the following ring-based formulation of the Syndrome Decoding Problem (SDP), known as QC-SDP.

Problem 1 (Quasi-Cyclic Syndrome Decoding Problem).
Given: $h, s \in \mathcal{R}$ and $w \in \mathbb{N}$.
Goal: find $e_0, e_1 \in \mathcal{R}$ with $wt(e_0) + wt(e_1) \leq w$ such that $e_0 + e_1 h = s$.

This was shown to be NP-complete in [2]. When $n_0 = 2$, it has been proved in [6] that random quasi-cyclic codes lie on the GV bound with overwhelming probability. Moreover, the impact of cyclicity on SDP has been studied, for example in [5], revealing no substantial gain.

3 A FRAMEWORK FOR SIGNATURES

There is a relatively recent approach that provides an easy way to construct efficient signature schemes based on any hard problem. The approach consists of successive reductions building on the original hard problem, first deriving a collision-resistant hash function f, then converting it into a one-time signature where the private key is a pair of integers (x, y), the public key is the pair $(f(x), f(y))$, and the signature of a message m is simply $mx + y$. The one-time signature can then be turned into an identification scheme by replacing m with a challenge c chosen by the verifier and letting y be the commitment (a distinct y is used in every run of the protocol). Finally, the identification scheme is transformed into a full-fledged signature scheme using the Fiat-Shamir transform. Proposals based on classical number theory problems such as RSA or discrete logarithm (see Okamoto [18]) are easy and intuitive to design. Lyubashevsky showed for the first time how to translate the framework to the lattice case, presenting in [14] an identification scheme which was then refined and updated in [15].

3.1 A Coding Theory Scenario

A first, direct translation of the framework to the case of code-based cryptography was proposed by Persichetti in [19]. The idea is for the scheme to rely on SDP, hence featuring a public matrix H, a secret x having weight below the GV bound and the public key $s_x = Hx^T$. Similarly to the lattice case, the final verification should include not only an algebraic formula consisting of H, the commitment Y and s_x, but also a check on the weight of the response z. Formally, one can see the syndrome computation as a hash function $f(x) = Hx^T$, which is is preimage-resistant provided that the weight of x is small. From now on, we will denote this function as $synd_H(x)$. It follows that the scheme is subject to the additional constraint that the random element y and the challenge c should be chosen such that $\mathrm{wt}(z) \leq w$, where w is the value of the GV distance. This means that c can only be an element of $\mathbb{F}_q$ and that x and y must satisfy $\mathrm{wt}(x) = \gamma_1 w, \mathrm{wt}(y) = \gamma_2 w$, for certain constants $\gamma_1, \gamma_2 \leq 1$ such that $\gamma_1 + \gamma_2 = 1$. Due to space limitations, we refer to [19] for a detailed description of the scheme.

Vulnerability from Multiple Signatures. Unfortunately, if used to sign multiple messages, this simple proposal is vulnerable to an attacker who tries to learn the secret. In fact, if an attacker can obtain a polynomial number of signatures, it could store the corresponding values of z and c and then compute $z' = c^{-1}y + x$: this is always possible, since c is a field element and is non-zero. Now, the vector $y' = c^{-1}y$ is randomly generated and has low weight, so each of its coordinates is biased towards 0. Therefore, a simple statistical analysis will eventually reveal all the positions of x. The problem seems to come from the scheme metric itself. In fact, c is constrained to be a field element (to fit the verification equation) but doesn't alter the weight of x, and so the low-weight vector y that is added is not enough to properly hide the secret support.

4 THE NEW SCHEME

The core of our idea is to use quasi-cyclic codes in the framework that we have described above. The use of quasi-cyclic codes in cryptography is not a novelty: these have been proposed before in the context of encryption (e.g. [2]). Their originally suggested use (i.e. with GRS codes) was cryptanalyzed in [8] and it is thus not recommended, but other variants based on LDPC and MDPC codes are still considered safe. In both cases, the issue is that introducing the extra algebraic structure can compromise the secrecy of the private matrix used for decoding. A big advantage of our proposal is that this issue does not apply. In fact, since there is no decoding involved, an entirely random code can be used, and the code itself is public, so there is no private matrix to hide.

As far as signature schemes go, Gaborit and Girault in [10] propose a variant of Stern's ID scheme that uses quasi-cyclic codes (called "double-circulant" by the authors). While this proves to be more efficient than the classical Stern scheme, the protocol still features the same flaw, i.e. a non-trivial cheating probability. This leads to the necessity of repeating the protocol several times, with an obvious impact on the efficiency of the scheme.

In our setting, we use 2-quasi-cyclic codes where words are vectors in $\mathcal{R} \times \mathcal{R}$. For a word $x = (x_0, x_1)$, the syndrome function associated to $h \in \mathcal{R}$ is defined as $synd_h(x) = x_0 + x_1 h$, following the notation that takes a parity-check matrix in systematic form (and hence defined by h) as in Problem 1. For a general formulation, we will use the notation $\mathcal{D}_1$ and $\mathcal{D}_2$ to indicate the distributions that sample uniformly at random vectors of $\mathcal{R} \times \mathcal{R}$ having weight respectively less or equal to $w_1 = \gamma_1 w$ and $w_2 = \gamma_2 w$. Our signature scheme is presented below. The scheme uses a hash function $\mathcal{H}$ that outputs bit strings of fixed weight δ, which is one of the system parameters.

KeyGen

Input: parameters $p, \delta, w_1, w_2 \in \mathbb{N}$ and a vector $h \in \mathcal{R}$.

(1) Sample $x \xleftarrow{\$} \mathcal{D}_1$.
(2) The signing key is x.
(3) The verification key is $s_x = synd_h(x)$.

Sign

Input: a message m and the signing key x.

(1) Sample $y \xleftarrow{\$} \mathcal{D}_2$.
(2) Compute $s_y = synd_h(y)$.
(3) Compute $c = \mathcal{H}(m, s_y)$.
(4) Compute $z = cx + y$.
(5) The signature is $\sigma = (c, z)$.

Ver

Input: a message m, a signature σ and the verification key s_x.

(1) Compute $s_z = synd_h(z)$.
(2) Use the verification key to compute $v = cs_x + s_z$.
(3) Compute $c' = \mathcal{H}(m, v)$.
(4) Accept if $c' = c$ and $\mathrm{wt}(z) \leq w$.
(5) Else, reject.

Like before, we have a constraint on the weight of the response vector z: in this case $w \leq \delta w_1 + w_2$ since c is no longer a constant. Then w is required to be below the GV bound to ensure that the response z is the unique solution to the corresponding QC-SDP instance. This is a consequence of the security requirements, as we will see next.

To conclude, note that it is easy to check that an honest verifier always gets accepted. In fact, in an honest run of the protocol, then $v = cs_x + s_z = c \cdot synd_h(x) + synd_h(z)$. Due to the transitivity of the syndrome computation, this is the same as $synd_h(cx + z) = synd_h(y) = s_y$. Therefore $c' = \mathcal{H}(m, v) = \mathcal{H}(m, s_y) = c$ and the verification is passed.

5 SECURITY

We begin by showing the impossibility of multi-signing.

Procedure 1. *Start by obtaining a polynomial number ℓ of signatures, i.e. pairs $(c^{(i)}, z^{(i)})$ for $i, = 1, \ldots, \ell$. For each pair, $c^{(i)}$ is chosen uniformly at random among the vectors of weight δ, and $z^{(i)} = c^{(i)}x + y^{(i)}$ where $y^{(i)}$ is also chosen uniformly at random (sampled from $\mathcal{D}_2$). For each i, write $c^{(i)} = X^{i_1} + \cdots + X^{i_\delta}$, that is, as a polynomial of weight δ in $\mathcal{R}$. Then calculate*

$$\begin{aligned} z^{(i,j)} &= X^{-i_j} z^{(i)} \pmod{X^p - 1} \\ &= X^{-i_j}(c^{(i)}x + y^{(i)}) \pmod{X^p - 1} \\ &= (1 + \sum_{k \neq j, k \in \{1,\ldots,\delta\}} X^{i_k - i_j})x + X^{-i_j} y^{(i)} \pmod{X^p - 1} \\ &= x + \sum_{k \neq j, k \in \{1,\ldots,\delta\}} x^{(i,j)} + y^{(i,j)} \pmod{X^p - 1} \end{aligned}$$

for $x^{(i,j)} = X^{i_k - i_j} x \pmod{X^p - 1}$, $y^{(i,j)} = X^{-i_j} y^{(i)} \pmod{X^p - 1}$.

Since $x^{(i,j)}$ is just a shift of x and $y^{(i,j)}$ is just a shift of $y^{(i)}$, and their support will likely have little to no intersection with the support of x (due to the weight of the vectors), it is possible to reveal the support of x simply by looking at the bits that belong to the support of a large enough number of $z^{(i,j)}$.

Note that the above procedure is in fact a refinement of the simple statistical analysis attack encountered before: in both cases, the problem is that the weight of the vectors is simply too low to properly mask the private vector. It is then clear that it is impossible to sign multiple times and preserve security. It follows that our scheme only achieves one-time security. To prove the one-time security of our scheme, we follow the paradigm for a generic one-time signature scheme of Pointcheval and Stern, which was already employed in the code-based setting in [1]. Details about security are omitted due to space limitations, hence we will present only our final result, and we refer again the reader to the full version of this paper for an extensive treatment.

Theorem 5.1. *Let $\mathcal{A}$ be a polynomial-time 1-EUF-CMA adversary for the signature scheme with parameters p, δ, w_1, w_2, running in time T and performing at most q random oracle queries. Let the probability of success of $\mathcal{A}$ be $\varepsilon \geq 20(q+1)/2^\lambda$. Then the QC-SDP problem with parameters $n = 2p$, $w = \delta w_1 + w_2$ can be solved in time $T' \leq 120686 \ell q T / \varepsilon$.*

Proof. The proof uses the well-known Forking Lemma, in an iterative way, to produce ℓ distinct, valid signatures. The thesis is immediately obtained by using these with Procedure 1. □

6 PERFORMANCE AND COMPARISON

We start by recalling the main components our scheme. The public data consists of the vector h (of length p) and the syndrome s_x (also of length p), for a total of $2p$ bits. The signature, on the other hand, is given by the challenge string c and the response z. In our scheme, this corresponds respectively to a vector of length p and a vector of length $2p$. It is possible to greatly reduce this size thanks to a storing technique [21] which allows to represent low-weight vectors in a compact manner. Namely, a binary vector of length n and weight w is represented as an index, plus an indication of the actual vector weight, for a total of $\log \binom{n}{w} + \log(w)$. Note that in our case this applies to both c and z.

We now provide some parameters for the codes in our scheme. These are normally evaluated with respect to general decoding algorithms such as Information-Set Decoding [3, 16, 17, 20, 22]: the amount of security bits is indicated in the column "Security".

Table 2: Parameters (all sizes in bits).

p	w_1	w_2	δ	Security (λ)	Public Data	Signature Size
4801	90	100	10	80	9602	4736
9857	150	200	12	128	19714	9475
3072	85	85	7	80	6144	3160
6272	125	125	10	128	12544	6368

The first two rows report well-known parameters suggested in the literature for QC-MDPC codes; however, since our codes do not need to be decodable, we are able to slightly increase the number of errors introduced. The last two rows, instead, are parameters chosen ad hoc, in order to optimize performance.

6.1 Comparison

We compare our scheme to other code-based proposals that are one-time secure, such as [1] and [11]. Both of these schemes follow the KKS approach [13], and therefore come with some potential security concerns. For simplicity, we will refer to [1] as BMS and to [11] as GS. Note that the latter comes in two variants, which use respectively quasi-cyclic codes, and a newly-introduced class of codes called "quadratic double-circulant" by the authors. All the parameters and sizes (in bits) are reported in the following table, and correspond to a security level of 2^{80}.

Table 3: Code-based one-time signature schemes.

	BMS	GS 1	GS 2	Our Scheme
Public Data	930080	75000	17000	6144
Signature Size	3739	18900	7000	3160

It is immediate to notice that our scheme presents the smallest amount of public data (public key plus any additional public information) and the smallest signature size. To be fair, the BMS scheme employs the same indexing trick used in this work, while this is not the case for the other scheme. Since the signature of the GS schemes also includes a low-weight vector, we expect that it would be possible to apply the same technique to the GS schemes as well, with the obvious reduction in size. We did not compute this explicitly but it is plausible to assume it would be very close to that of our scheme. Nevertheless, the size of the public data remains much larger even in the most aggressive of the two variants (GS 2).

6.2 Implementation

To confirm the practicality of our scheme, we have developed a simple implementation in C. The implementation is a straightforward translation to C with the addition of the steps for generating public and private keys. The hash function used was SHA-256. We ran the protocol on a small microprocessor, namely a 580 MHz single-core MIPS 24KEc. The choice of this microprocessor was made based on the usage of it, since this type of microprocessor is commonly used in the Internet of Things (IoT) applications. The measurements are reported below.

Table 4: Implementation Results.

p	w_1	w_2	δ	KeyGen (sgk/vk)	Sign	Ver
4801	90	100	10	0.061ms / 22.754ms	89.665ms	22.569ms
9857	150	200	12	0.169ms / 104.655ms	374.206ms	99.492ms
3072	85	85	7	0.052ms / 14.017ms	35.150ms	14.271ms
6272	125	125	10	0.116ms / 67.972ms	150.063ms	42.957ms

Note that key generation is dominated by the syndrome computation necessary to obtain the verification key, while sampling the signing key has a negligible cost. The signing operation is the most expensive, which makes sense, while the verification is of the same order of magnitude as the key generation. Both signing and verification algorithm are relatively fast but could be sped up even further, since the hash function used was, at the time the measurements were taken, not optimized to run in such a small device.

REFERENCES

[1] P. S. L. M. Barreto, R. Misoczki, and M. A. Simplício Jr. One-time signature scheme from syndrome decoding over generic error-correcting codes. *Journal of Systems and Software*, 84(2):198–204, 2011.

[2] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. Reducing Key Length of the McEliece Cryptosystem. In B. Preneel, editor, *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 77–97. Springer, 2009.

[3] D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the mceliece cryptosystem. In J. Buchmann and J. Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008.

[4] P.-L. Cayrel, P. Véron, and S. M. El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 2010.

[5] C. Chabot and M. Legeay. Using automorphisms group for decoding. In *12th internaional workshop on Algebraic and Combinatorial Coding Theory (ACCT 2010)*.

[6] C. L. Chen, W. W. Peterson, and E. J. Weldon. Some results on quasi-cyclic codes. *Information and Control*, 15(5):407–423, 1969.

[7] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a mceliece-based digital signature scheme. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer, 2001.

[8] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2010.

[9] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[10] P. Gaborit and M. Girault. Lightweight code-based identification and signature. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 191–195. IEEE, 2007.

[11] P. Gaborit and J. Schrek. Efficient code-based one-time signature from automorphism groups with syndrome compatibility. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 1982–1986. IEEE, 2012.

[12] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[13] G. Kabatianskii, E. Krouk, and B. J. M. Smeets. A digital signature scheme based on random error-correcting codes. In M. Darnell, editor, *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 161–167. Springer, 1997.

[14] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

[15] V. Lyubashevsky. Lattice signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 738–755. Springer, 2012.

[16] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $O(2^{0.054n})$. In D. H. Lee and X. Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011.

[17] A. May and I. Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Advances in Cryptology - EUROCRYPT* 2015, Lecture Notes in Computer Science. Springer, 2015.

[18] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer, 1992.

[19] E. Persichetti. *Improving the Efficiency of Code-Based Cryptography*. PhD thesis, University of Auckland, 2012.

[20] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, pages 5–9, 1962.

[21] F. Ruskey. Combinatorial generation. *Preliminary working draft. University of Victoria, Victoria, BC, Canada*, 11:20, 2003.

[22] J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

[23] J. Stern. A new identification scheme based on syndrome decoding. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993.

[24] J. Stern. Designing identification schemes with keys of short size. In Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 164–173. Springer, 1994.

[25] P. Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.