Computer Science Journal of Moldova, vol.**16**, no.1(46), 2008

Private Key Extension of Polly Cracker Cryptosystems

Nina Taslaman

Abstract

In 1993 Koblitz and Fellows proposed a public key cryptosystem, *Polly Cracker*, based on the problem of solving multivariate systems of polynomial equations, which was soon generalized to a Gröbner basis formulation. Since then a handful of improvements of this construction has been proposed.

In this paper it is suggested that security, and possibly efficiency, of any Polly Cracker-type cryptosystem could be increased by altering the premises regarding private - and public information.

1 Introduction

In 1993, Koblitz and Fellows [1] proposed a public key cryptosystem, *Polly Cracker*, based on the NP-complete problem of solving multivariate systems of polynomial equations over a finite field. This was immediately generalized to a Gröbner basis formulation, where the problem of solving polynomial equations was replaced by the EXPSPACE-complete problem of computing a Gröbner basis for an ideal. Using some general NP - or EXPSPACE-complete problem as the basis for a public key cryptosystem was a daring move, since the failure of Merkle and Hellman's knapsack-based cryptosystem from 1978 [4] had resulted in high scepticism among cryptographers regarding this type of construction. Indeed, a title like *Why you cannot even hope to use Gröbner Bases in Public Key Cryptography* [3] suggests it met a harsh response. The main criticism against the idea was single-break attacks (i.e. individual-message recovery) based on linear algebra.

However, over the years a plethora of possible countermeasures against these attacks and others has been proposed, as well as different modifications to improve and generalize the initial idea - the most general version as of now seeming to be Ackermann and Kreutzer's generalization to module Gröbner bases over general monoid rings, which allows commonly used public key schemes such as RSA and ElGamal to be formulated as special cases [9].

Rather than continuing in this direction of generalizing the setting, V. Ufnarovski suggested author to investigate altering the rules for private and public information in the Polly Cracker setup. This is the subject of this paper.

To introduce the actors: Alice - intended receiver of secret messages, Bob - sender of such messages, and Eve - enemy, who tries to recover Bob's messages. Messages are restricted to some message space M and encrypted by Bob using some encryption function $F : M \to C$ into ciphertext space C. In a public-key cryptosystem (Williamson 1974 [5], Diffie and Hellman 1976 [6]) there may be many Bob's but only one Alice, i.e. F is publicly known (the *public key*) and anyone may encrypt messages, but (hopefully) only Alice can decipher them. This requires F to be a *trapdoor one-way function*, i.e. while encryption $F(m) = c$ may be computed in polynomial time, the decryption $F^{-1}(c) = m$ may not - except for someone (Alice) knowing some additional trapdoor information which simplifies the computation (the *private key*). As for Eve's part of the game, one distinguishes between *total break attacks*, in which she tries to find the secret key (or some equivalent information) so that she may decrypt any future ciphertext, and *single break attacks*, aimed at decrypting specific individual messages. The basic assumption is always that Eve has access to any encrypted message sent by Bob. One also has to consider the situation that she has temporary access to some decryption black box (e.g. in the form of a compiled decryption program), which she may use to decrypt any finite number of ciphertexts of her choosing. This is the scenario for a *chosen-ciphertext attack*, where Eve's goal is to use this information for a total break attack.

1.1 The Polly Cracker Public Key System

Let $\mathbb{F}_q[X]$ be the set of multivariate polynomials over a finite field $\mathbb{F}_q$ generated by the alphabet $X = \{x_1, ..., x_n\}$. Given a subset F of polynomials, let $\langle F \rangle$ denote the ideal they generate over $\mathbb{F}_q[X]$. Also, given a Gröbner basis $G \subset \mathbb{F}_q[X]$, under some monomial ordering $\preceq$, and a polynomial $f \in \mathbb{F}_q[X]$, let $\bar{f}$ denote the normal form of f over $\langle G \rangle$ with respect to $\preceq$, i.e. $\bar{f} := r_G(f)$ is the unique remainder of f over G under the given monomial ordering. The Gröbner basis version of Polly Cracker may then be described like so:

Cryptosystem 1.1 (Polly Cracker).

Key generation To set up the system, Alice chooses a Gröbner basis $G \subset \mathbb{F}_q[X]$ under some monomial ordering $\preceq$ and selects a finite subset $P \subset \langle G \rangle$ of the corresponding ideal.

Private Key: G Public Key: P

Message space A subset of all G-normal forms:

$$M \subset \{ \ \bar{f} \mid f \in \mathbb{F}_q[X] \ \}$$

Encryption Bob encrypts a message $m \in M$ by choosing some $p \in \langle P \rangle$ and computing the ciphertext

$$c := m + p \in m + \langle G \rangle$$

Decryption Alice decrypts c by computing its normal form over $\langle G \rangle$:

$$\bar{c} = r_G(c) = r_G(m) + r_G(p) = m + 0$$

1.2 Main Attacks

A total break attack on this cryptosystem generally amounts to computing an equivalent Gröbner basis G' for the public key ideal $\langle P \rangle$ - this would be an equivalent secret key. The general problem of computing a Gröbner basis for a given ideal is NP-complete (see e.g. [7]), and Alice may choose her P and $\preceq$ from some class of known hard instances, for example by encoding well-studied problems from logic, to ensure giving Eve (the attacker) a hard time here.

Now, if Eve does not succeed in the above she could always try to exploit some possible weakness in Bob's choice of $p \in \langle P \rangle$, letting her decipher at least some of his messages. The most severe criticism against Polly Cracker has been its vulnerability to such single break attacks based on linear algebra, mentioned already in Fellows and Koblitz's original paper [1]. With public key $P = \{p_1, ..., p_s\}$, Bob's p will have the form $p = \sum_{i=1}^{s} h_i p_i$ for some ephemeral polynomials h_i. The main idea is then to consider

$$c = m + \sum_{i=1}^{s} h_i p_i \tag{1}$$

as a linear system of equations, whose unknowns are the coefficients of the polynomials h_i's and m. By guessing the support of these, the linear system might be solvable by usual Gaussian elimination, retrieving m. The countermeasure here is for Alice to choose the setting parameters so as to ensure infeasible system sizes (there is a security/efficiency-tradeoff here), and for Bob to choose his h_i's so as to ensure a certain amount of cancellation in the sum. This calls for quite clever constructions.

1.3 Efficiency Issues

The main problem for implementing Polly Cracker instances stems from the above mentioned security/efficiency-tradeoff. In particular, the so called *message expansion* is an issue here: a message m will be encrypted into a ciphertext polynomial c of, most likely, larger support, so even though *supp*(m) may be as small as a single constant term, *supp*(c) may be very big if parameter sizes are not properly restricted, implying issues in storage, transfer and decryption. For example, in [2] Koblitz presents a study-example of a Polly Cracker instance (the Graph Perfect Code Instance) based on a perfect code problem from graph theory, and for sufficient security suggests using a polynomial ring with 500 indeterminates x_i. However, even narrowing it down to 200 one gets ciphertexts of about 60'000 monomials for this instance (see [8]). All serious attempts at practical, implementable Polly Cracker instances have to deal with this issue, which tends to make them somewhat technical.

2 Related Work

In [10] (2004), Levy-dit-Vehel and Perret describe how to construct Polly Cracker instances based on 3-SAT problems from logic, i.e. so that a total break attack may be P-reduced to some well-studied hard 3-SAT instance, while at the same time providing resistance against the classical linear algebra attacks. The latter is achieved by the use of an elaborate generating algorithm for $p \in \langle P \rangle$, together with suggested parameter sizes resulting in a message expansion of about 1500 terms, which is at least manageable but still not suitable for practical use.

The efficiency issue is addressed more directly in [11] (2002), where Ly presents a cleverly constructed, however somewhat technical, modification of Polly Cracker called Polly Two. This cryptosystem can be viewed in three different polynomial settings via a ring homomorphism: domain- goal- or quotient ring, each setting providing security in its own way and simultaneously taking care of the efficiency/sequrity trade-off. In the goal-ring setting this cryptosystem reduces to a Polly Cracker instance with very large parameter sizes, thus handling the linear algebra attacks. Legal users operate in the domain ring where parameter sizes are quite small, with a message expansion of less than 100 terms. This would be acceptable for practical use, however setting up concrete instances seems to be somewhat difficult (e.g. finding a suitable homomorphism).

In [12] (2004), T. Rai generalizes Polly Cracker to noncommutative polynomial rings, inspiration being that this allows ideals for which no finite Gröbner bases exist. The idea here is for Alice to take a secret key Gröbner basis G, finite as usual, but with a public key subset $P \subset G$ so that no finite Gröbner basis exists for $\langle P \rangle$. This means that Eve cannot even theoretically succeed in the usual total break attack. Another benefit comes from the use of two-sided ideals, leading to quadratic (rather than linear) systems of coefficients in the single break attacks. Unfortunately, finding suitable ideals for concrete instances turns out to be a challenging task. Also, no experimental data is provided, so it is unclear how efficient instances of this system would be.

Going further along the generalizing path, Ackermann and Kreuzer in [9] take the Polly Cracker scheme all the way up to a setting of modules (generalizing the ideals in Polly Cracker) over general monoid rings (generalizing the standard polynomial rings). This could be a promising framework for future cryptosystems (no such instances are provided), but even in its abstract formulation it is of direct interest since most well-known public key schemes seem to let themselves be formulated as special cases, e.g. RSA, ElGamal and even recent attempts at group-based public key schemes.

3 Extending The Private Key in Polly Cracker

Studying the Polly Cracker construction (Cryptosystem 1.1), we make the following observations:

1. The monomial ordering $\preceq$ used is seemingly assumed to be a public domain parameter - at least the advantages of keeping it private is, to our knowledge, never pointed out. The idea here is the following:

   > Alice could choose a Gröbner basis G under some ordering $\preceq$ so that $\langle G\rangle$-normal words with respect to $\preceq$ are not necessarily $\langle G\rangle$-normal with respect to other orderings.

   This would imply that even if Eve managed to find some Gröbner basis $\tilde{G}$ for $\langle P\rangle$, unless she guesses the correct monomial ordering, she cannot expect messages to be preserved in an attempted decryption, i.e. it might be that

$$r_{\tilde{G}}(c) = r_{\tilde{G}}(m) \neq r_G(m) = m$$

2. The public setting for Polly Cracker is a polynomial ring over some finite field $\mathbb{F}_q$. It is never motivated why the cardinality of this field should be public information. In fact, Bob could encrypt messages perfectly well in $\mathbb{Z}[X]$, with Alice taking the ciphertext ($mod\ p$) before proceeding as usual with decryption, if we just require the coefficients of messages to be bounded so that they are not destroyed by the ($mod\,p$) computation.

While the idea of private monomial ordering works with the usual Polly Cracker scheme, keeping the field cardinality private requires some adjustments of the scheme.

3.1 Polly Goes Private - With p

To concretize these ideas, let us first for simplicity of discussion consider the case $\mathbb{F}_q = \mathbb{Z}_p$ for some large prime number p. For a set of polynomials $F \subset \mathbb{Z}_p[X]$, let $\langle F\rangle_p$ denote the usual ideal they generate in $\mathbb{Z}_p[X]$, and let $\langle F\rangle_{\mathbb{Z}}$ denote the ideal F generates when lifted to $\mathbb{Z}[X]$, i.e.

$$\langle F\rangle_{\mathbb{Z}} := \{\sum_{f\in F} f h_f \mid h_f \in \mathbb{Z}[X]\}$$

Note that

$$\langle F\rangle_{\mathbb{Z}}\ (mod\,p) = \langle F\rangle_p \tag{2}$$

Cryptosystem 3.1 (Polly Cracker with Private $\preceq$ and p).

Key generation Alice chooses some big prime p, a positive integer $q < p$, a finite Gröbner basis $G \subset \mathbb{Z}_p[X]$ under some monomial ordering $\preceq$, and a finite subset $P \subset \langle G\rangle_{\mathbb{Z}}$.

Private Key: p, G, $\preceq$ Public Key: P

Message space M: G-normal forms under $\preceq$ in $\mathbb{Z}_p[X]$ with coefficients bounded by q.

Encryption Bob chooses $f \in \langle P \rangle_{\mathbb{Z}}$ and encrypts a message $m \in M$ into the ciphertext

$$c := m + f \in \mathbb{Z}[X]$$

Decryption Alice decrypts c by first computing

$$c' = c \ (mod\, p) = m + f_p \in \mathbb{Z}_p[X]$$

where $f_p := f \ (mod\, p)$ and then

$$\bar{c'} = r_G(c') = r_G(m) + r_G(f_p) = m + 0$$

Decryption follows from (2):

$$f \in \langle P \rangle_{\mathbb{Z}} \ \ \Rightarrow \ \ f \ (mod\, p) \in \langle P \rangle_p \subset \langle G \rangle_p$$

Before proceeding with the case of higher prime-power cardinality, let us first discuss the effects of this private key alteration.

3.1.1 Security gain

The main idea of keeping p private is that it blows up the complexity of a total break attack. As before, this attack amounts to finding a Gröbner basis (under some lucky monomial ordering) for $\langle P \rangle_p$. While this can be made hard even when p is known, without this knowledge Eve could at best try searching through primes $p' > q$, and for each try finding a Gröbner basis for $\langle P \rangle_{p'}$.

Also, forcing users to compute over $\mathbb{Z}[X]$, rather than $K[X]$ for some field K, Eve cannot use scalar inverses in her attacks. Since Gaussian elimination without using scalar inverses leads to intermediate coefficient swell, this means that linear algebra attacks grow more costly.

3.1.2 Efficiency possibilities

The decryption procedure now consists of two steps: first a modulo operation, which is fast, and then the usual reduction, which may be costly. Alice has a possibility to speed up the decryption procedure here by choosing some public key polynomials $p_i \equiv 0\ (mod\, p)$, so that much of the ciphertext is simplified in the first (fast) decryption step. While tempting for very efficient decryption, Alice should not take every $p_i \equiv 0\ (mod\, p)$, however, since this would make p a common factor of all public-key coefficients, which could be detected by Eve.

3.1.3 Issues and countermeasures

By limiting message coefficients to $q < p$, there is a trade-off between the size of the message space and the additional security provided by keeping p secret. However, if q and p are large enough, this should not be a major concern.

A more serious effect is that, since Bob encrypts over $\mathbb{Z}[X]$, the coefficients of the ciphertext may grow big, which can be cumbersome. To limit this effect he should not choose ephemeral key polynomials with too big coefficients. The Chinese remainder theorem could also be used for more efficient transmission:

With α the largest coefficient of a ciphertext polynomial c, Bob multiplies relatively prime numbers n_i, of manageable size, so that the product $N := n_1 \cdots n_r \geq \alpha$. He then computes

$$\begin{cases} c_1 = c & (mod\, n_1) \\ \quad ... & \\ c_r = c & (mod\, n_r) \end{cases} \tag{3}$$

and sends the ciphertext tuple

$$C := \{(c_1, ..., c_r), (n_1, ..., n_r)\}$$

Here coefficients of the c_i's are bounded by $max\{n_1, ..., n_r\}$. Alice then uses the Chinese remainder theorem to solve (3), recovering $c\ (mod\, N) = c$ with full coefficients, and she may proceed as before.

Note, however, that while coefficient sizes may be controlled by this method, we have to pay in the number r of ciphertext polynomials.

3.1.4 Chosen-ciphertext attack

In a private letter, Rai suggests a chosen-ciphertext attack aimed at finding our secret p: Eve could e.g. enumerate primes $q_i > q$ and encrypt fake messages of the form

$$\tilde{m} = q_1 m_1 + ... + q_k m_k$$

where each m_i is a monomial in the message space. If it would happen that some $q_i = p$, the corresponding term would decrypt to zero and the decryption black box she has temporary access to would return $\tilde{m} - q_i m_i$, revealing $p = q_i$.

Note that such a fake message after decryption would contain some coefficients $q_j > q$, which was not allowed in the message space. Hence, to avoid this attack, the decryption black box should be set to detect any such fake ciphertexts (decrypting to terms with coefficients larger than q) and return an error message if that happens.

3.2 Polly Goes Private - With p^n

Now suppose $\mathbb{F}_q = \mathbb{F}_{p^n}$ for some prime p and $n > 1$, and let α denote a generating element for this field via some primitive degree-n-polynomial in $\mathbb{Z}_p[\alpha]$. We use α-power notation as default for nonzero field elements. Let us define a homomorphism from the ring of univariate polynomials $f(s)$ over $\mathbb{Z}$ into $\mathbb{F}_{p^n}$ by

$$\tilde{\varphi} : \mathbb{Z}[s] \to \mathbb{F}_{p^n}; \quad s \mapsto \alpha$$

and extend it to a homomorphism from $\mathbb{Z}[s][X]$ into $\mathbb{F}_{p^n}[X]$ as:

$$\varphi : \mathbb{Z}[s][X] \to \mathbb{F}_{p^n}[X]; \quad f(s)w \mapsto \tilde{\varphi}(f)w \tag{4}$$

where w denotes a word with letters from X. This φ will be used by Alice to translate Bob's messages in $\mathbb{Z}[s][X]$ into the ordinary Polly Cracker setting $\mathbb{F}_{p^n}[X]$. We will need some notation here in order to recognize corresponding key polynomials in these two settings.

Given $f = \sum \alpha^k w_k$ in $\mathbb{F}_{p^n}[X]$, let f_s denote the polynomial obtained in $\mathbb{Z}[s][X]$ by simply replacing every α by s. Then, corresponding to the definitions in the prime-cardinality case, for $F \subset \mathbb{F}_{p^n}[X]$ let $\langle F\rangle_{p^n}$ be the usual ideal generated by F over $\mathbb{F}_{p^n}[X]$, i.e.

$$\langle F\rangle_{p^n} := \{\sum_{f\in F} fg_f \mid g_f \in \mathbb{F}_{p^n}[X]\}$$

and let

$$\langle F\rangle_{\mathbb{Z}[s]} := \{\sum_{f\in F} f_s h_f \mid h_f \in \mathbb{Z}[s][X]\}$$

be the ideal generated by the corresponding polynomials f_s over $\mathbb{Z}[s][X]$. Since

$$\varphi(f_s h_f) = \varphi(f_s)\varphi(h_f) = f\varphi(h_f)$$

we have

$$f \in \langle F\rangle_{\mathbb{Z}[s]} \quad\Rightarrow\quad \varphi(f) \in \langle F\rangle_{p^n} \tag{5}$$

Now, Alice may keep p and n secret while letting Bob compute over $\mathbb{Z}[s][X]$. Using φ she may then translate his ciphertext into a standard Polly Cracker ciphertext in $\mathbb{F}_{p^n}[X]$. By 5, this works if the message space is restricted properly. The details are as follows:

Cryptosystem 3.2 (Polly Cracker with Private $\preceq$ and p^n).

Key generation Alice chooses a prime number p, some $n > 1$, a finite Gröbner basis $G \subset \mathbb{F}_{p^n}[X]$ under some monomial ordering $\preceq$, a finite subset $P \subset_R \langle G\rangle_{\mathbb{Z}[s]}$ and some $r < p^n - 1$.

Private Key: $\mathbb{F}_{p^n}$, G, $\preceq$ Public Key: P

Message space Linear combinations of G-normal words $w_i \in \mathbb{F}_{p^n}[X]$ with coefficients s^k where $k < r$, i.e.

$$M = \{\sum s^{k_i} w_i \mid k_i \leq r,\ r_G(w_i) = w_i\}$$

Encryption Bob chooses $f \in \langle P\rangle_{\mathbb{Z}[s]}$ and encrypts a message $m = \sum s^{k_i} w_i \in M$ into the ciphertext

$$c := m + f \in \mathbb{Z}[s][X]$$

Decryption Alice decrypts c as

$$r_G(\varphi(c)) = r_G(m_\alpha + \varphi(f)) = m_\alpha + 0$$

where $m_\alpha = \sum \alpha^{k_i} w_i$ is the message m only with the symbol s replaced by α.

Here decryption follows from 5:

$$f \in \langle P \rangle_{\mathbb{Z}[s]} \subset \langle G \rangle_{\mathbb{Z}[s]} \quad \Rightarrow \quad \varphi(f) \in \langle G \rangle_{p^n}$$

Note that the message is preserved in two steps: First it is preserved by φ since its coefficients are of form s^k for $k < q^n - 1$ (so there is no modulo-effect in the exponent), and then it is preserved in reduction over G, as usual for Polly Cracker, being a normal form.

In this description we have, for clarity, used the different symbols s and α to distinguish Bob's computations over $\mathbb{Z}[s][X]$ from field computations. Of course we might as well let Bob use the same symbol α and compute over $\mathbb{Z}[\alpha][X]$ - the important thing is that he is not able to interpret α as the field element in $\mathbb{F}_{p^n}$.

Example 3.1 (Toy Example). For demonstration, we give a very small example in $\mathbb{F}_{2^3}[x, y]$. A translation table for power/polynomial representation of the field elements in $\mathbb{F}_{2^3}$ is given by:

α^k	$r_k(\alpha)$	α^k	$r_k(\alpha)$
-	0	α^3	$\alpha + 1$
1	1	α^4	$\alpha^2 + \alpha$
α	α	α^5	$\alpha^2 + \alpha + 1$
α^2	α^2	α^6	$\alpha^2 + 1$

Key generation Take the Gröbner basis

$$G = \{x - \alpha^5, y - \alpha^2\} \in \mathbb{F}_{2^3}[x, y]$$

and preliminary public key polynomials

$$\hat{p}_1 = x^2 + \alpha xy + 1, \quad \hat{p}_2 = \alpha^2 xy + \alpha y^2 + \alpha^3$$

Over $\mathbb{F}_{2^3}$ we have $\hat{p}_1(\alpha^5, \alpha^2) = \hat{p}_2(\alpha^5, \alpha^2) = 0$, so

$$\hat{p}_1, \hat{p}_2 \in \langle G \rangle_{p^n}$$

We multiply these by some polynomials in $\mathbb{Z}[\alpha][x, y]$ to form public key polynomials $p_1, p_2 \in \langle G \rangle_{\mathbb{Z}[\alpha]}$, for example:

$$p_1 = \hat{p}_1 \cdot (5\alpha^7 x + 1) = 5\alpha^7 x^3 + 5\alpha^8 x^2 y + x^2 + \alpha xy + 5\alpha^7 x + 1$$

$$p_2 = \hat{p}_2 \cdot (4\alpha^2 y - \alpha) = 4\alpha^4 xy^2 + 4\alpha^3 y^3 - \alpha^3 xy - \alpha^2 y^2 + \alpha^5 y - \alpha^4$$

For message restriction we choose $r = 6 < 2^3 - 1$.

Private Key: $\mathbb{F}_{2^3}$, $G = \{\ x - \alpha^5,\ y - \alpha^2\ \}$

Public Key: $P = \{\ p_1,\ p_2\ \}$ from above

Message space G-normal forms in this case are just constants:

$$M = \{\alpha^k \mid k \leq 6\}$$

Encryption Suppose Bob wants to send us the message $m = \alpha^6$. He chooses ephemeral polynomials in $\mathbb{Z}[\alpha][x, y]$:

$$h_1 = 3y - \alpha, \quad h_2 = xy + \alpha^2$$

and computes the ciphertext in $\mathbb{Z}[\alpha][x, y]$:

$$c = m + p_1 h_1 + p_2 h_2 =$$

$$4\alpha^4 x^2 y^3 + 4\alpha^3 y^4 x - \alpha^3 x^2 y^2 - \alpha^2 xy^3 + (15\alpha^7 + 3)x^2 y +$$
$$(15\alpha^8 + 4\alpha^6 + 4\alpha^5 + 3\alpha)xy^2 + 4\alpha^5 y^3 - (5\alpha^8 + \alpha)x^2 -$$
$$(5\alpha^9 + \alpha^5 + \alpha^4 + \alpha^2)xy - \alpha^4 y^2 + (19\alpha^7 + 3)y - (5\alpha^8 + \alpha)$$

Note that Bob's choice of the last term α^2 in h_2 gives cancellation of the message $m = \alpha^6$ in c.

Decryption Upon receiving c as above, we first compute in $\mathbb{F}_{2^3}$ (using the translation table):

$$\begin{gathered}\varphi(c) = 0 + 0 + \alpha^3 x^2 y^2 + \alpha^2 x y^3 + (1+1)x^2 y + \\ (\alpha + 0 + 0 + \alpha) x y^2 + 0 + (\alpha + \alpha) x^2 + \\ (\alpha^2 + \alpha^5 + \alpha^4 + \alpha^2) x y + \alpha^4 y^2 + (1+1) y + (\alpha + \alpha) \\ = \alpha^3 x^2 y^2 + \alpha^2 x y^3 + x y + \alpha^4 y^2\end{gathered}$$

Then, with $G = \{x - \alpha^5, y - \alpha^2\}$ we have:

$$r_G(\varphi(c)) = \varphi(c)(\alpha^5, \alpha^2) = \alpha^3 + \alpha^6 + 1 + \alpha = \alpha^6 = m$$

$\lrcorner$

4 Conclusion

An extension of the private key in Polly Cracker has been suggested. In particular, an adjustment of the scheme to private field cardinality could be used to increase complexity of standard attacks (total- as well as single break), while at the same time providing means to control efficiency of decryption by introducing a fast preliminary decryption step before the usual reduction. This scheme adjustment is very simple in Polly Cracker instances over $\mathbb{Z}_p[X]$. The case of higher prime power coefficient fields requires a bit more theory, but in the end does not increase the complexity of the system. An issue that arises is the possible occurrence of large integer coefficients in the ciphertext. Modular techniques could be used to handle this effect.

It would remain to test these ideas on realistic Polly Cracker instances.

References

[1] M. Fellows, N. Koblitz, *Combinatorial cryptosystems galore!*, Finite fields: theory, applications and algorithms, Contemporary Mathematics Volume 168, 1994.

[2] N. Koblitz: *Algebraic aspects of cryptography*, Algorithms and Computation in mathematics, 3. Springer Verlag, 1998.

[3] B. Barkee, D. C. Can, J. Ecks, T. Moriarty, R.F. Ree: *Why you cannot even hope to use Gröbner Bases in Public Key Cryptography - An open letter to a scientist who failed and a challenge to those who have not yet failed*, Journal of Symbolic Computation, 18, pp. 497–501, 1994.

[4] R. Merkle, M. Hellman: *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Trans. Information Theory, 24(5), pp.525–530, 1978.

[5] M. J. Williamson: *Non-Secret Encryption Using a Finite Field*, 1974, `http://www.mirrors.wiretapped.net/security/info/reference/cesg-publications/History/secenc.pdf`.

[6] W. Diffie, M. Hellman: *New Directions in Cryptography*, IEEE Transactions on Information Theory, vol. IT-22, pp: 644–654, 1976.

[7] E. Mayr, A. Meyer, *The complexity of the word problems for commutative semigroups and polynomial ideal*, Adv. Math. Vol 46 no.3, pp. 305–329, 1982.

[8] D. Hofheintz, R. Steinwandt: *A "Differential" Attack on Polly Cracker*, IEEE International Symposium on Information Theory, Proceedings of ISIT 2002, p.211.

[9] P. Ackermann, M. Kreuzer: *Gröbner Basis Cryptosystems*, Universität Dortmund, 2006, `http://www.springerlink.com/content/174x321n05136859/fulltext.pdf`

[10] F. Levy-dit-Vehel, L. Perret, *A Polly Cracker system based on Satisfiability*, Progress in Computer Science and Applied Logic, Vol. 23, Birkhäuser, pp.177-192, 2004.

[11] L. V. Ly: *Polly Two - a public-key cryptosystem based on Polly Cracker*, Ruhr-Universität, 2002.

[12] T. S. Rai: *Infinite Gröbner Bases and Noncommutative Polly Cracker Cryptosystems*, Ph.D thesis, Virginia Polytechnic Institute and State University, 2004, `http://scholar.lib.vt.edu/theses/available/etd-03262004-082608/unrestricted/rai_etd.pdf`

Nina Taslaman,

E–mail: *ninus.af.quark@gmail.com*

Received February 18, 2008