BREAKING THE AKIYAMA-GOTO CRYPTOSYSTEM

by

Petar Ivanov & José Felipe Voloch

Abstract. — Akiyama and Goto have proposed a cryptosystem based on rational points on curves over function fields (stated in the equivalent form of sections of fibrations on surfaces). It is easy to construct a curve passing through a few given points, but finding the points, given only the curve, is hard. We show how to break their original cryptosystem by using algebraic points instead of rational points and discuss possibilities for changing their original system to create a secure one.

1. The cryptosystem of Akiyama and Goto

In this section we present the cryptosystem described by Akiyama and Goto in [**1**]. Let p be a prime number, $R = \mathbb{F}_p[t]$ be the polynomial ring over the prime field $\mathbb{F}_p$ and $K = \mathbb{F}_p(t)$ be the field of rational functions over $\mathbb{F}_p$. K is the field of fractions of R. Pick a polynomial in two variables $X(x, y) \in R[x, y]$, together with two points $U = (u_x, u_y) \in R^2$ and $V = (v_x, v_y) \in R^2$, such that $X(U) = X(V) = 0$. In other words, we take an algebraic curve over K together with 2 rational points on the curve. It is easy to find two points and a curve passing through them and we will show how below. On the other hand, if the curve is given (in terms of the polynomial $X(x, y)$), it is a hard mathematical problem to find rational points on it. This fact can be used to build the cryptosystem described in this section.

Keywords:Cryptosystem, Algebraic Curve, Rational Point.

1.1. Keys and key generation. — The secret key consists of the two points $U = (u_x(t), u_y(t))$ and $V = (v_x(t), v_y(t))$, such that either $\deg u_x \neq \deg v_x$ or $\deg u_y \neq \deg v_y$.

The public key consists of four things: the prime number p, the equation $X(x, y) = 0$, defining a curve through U and V, an integer l, which will serve as a lower bound for the degree of a monic irreducible polynomial $f \in R$ and an integer d, satisfying

$$d \geq \max\{\deg u_x, \deg u_y, \deg v_x, \deg v_y\}. \tag{1}$$

Let us write the equation of the curve as

$$X(x, y) = \sum_{i,j} c_{ij} x^i y^j = 0$$

and try to obtain the coefficients $c_{ij} \in R$ in such a way that U and V satisfy it. This means:

$$\sum_{i,j} c_{ij} u_x^i u_y^j = \sum_{i,j} c_{ij} v_x^i v_y^j = 0. \tag{2}$$

If we subtract the second sum from the first we get $\sum_{(i,j)\neq(0,0)} c_{ij}(u_x^i u_y^j - v_x^i v_y^j) = 0$, which can be written as

$$c_{10}(u_x - v_x) = - \sum_{(i,j)\neq(0,0),(1,0)} c_{ij}(u_x^i u_y^i - v_x^i v_y^i) = 0. \tag{3}$$

Now suppose that $(u_x - v_x)|(u_y - v_y)$. Then the right hand side of (3) is also divisible by $u_x - v_x$, because $u_x^i u_y^j - v_x^i v_y^j = (u_x^i - v_x^i)u_y^j + v_x^i(u_y^j - v_y^j)$. This suggest the following algorithm for choosing X:

1. For each pair of indices $(i, j) \neq (0, 0), (1, 0)$ pick a random element $c_{ij} \in R$.
2. Randomly choose elements λ_x, λ_y, v_x, v_y in R, such that $\lambda_x|\lambda_y$.
3. Compute $u_y = \lambda_y + v_y$ and $u_x = \lambda_x + v_x$.
4. Compute c_{10} from (3).
5. Compute c_{00} from (2) as $-c_{00} = \sum_{(i,j)\neq(0,0)} c_{ij} u_x^i v_u^j$.

1.2. Encryption and decryption. — Let $m \in R$ be the secret message that we want to encrypt and assume $\deg m < l$. When $p = 2$ we can encode any sequence of k bits as a polynomial of degree k in $\mathbb{F}_2[t]$. The assumptions mean that we need to encrypt the secret message by dividing it into blocks of at most l bits and encrypting each of them individually. The encryption goes like this:

1. Choose a random polynomial $s(x, y) \in R[x, y]$, which satisfies the following condition

$$(\deg_x s + \deg_y s)d + \deg_t s < l. \tag{4}$$

2. Choose another random polynomial $r(x, y) \in R[x, y]$ and a monic irreducible polynomial $f \in R$, such that

$$\deg_t f > l. \tag{5}$$

3. Compute the cypher polynomial $F(x, y) \in R[x, y]$ according to the formula

$$F = m + fs + Xr. \tag{6}$$

Now a person knowing the secret key, namely the points U and V, can easily decypher the encrypted polynomial F in the following way:

1. Evaluate F at U and V to get polynomials $h_1,\ h_2 \in R$.

$$h_1 = F(u_x, u_y) = m + fs(u_x, u_y)$$

$$h_2 = F(v_x, v_y) = m + fs(v_x, v_y).$$

2. Factor $h_1 - h_2$ and find f as the factor with largest degree.
3. Compute m as the remainder of h_1 when divided by f.

Note that indeed $f(t)$ is the highest degree factor of $h_1(t) - h_2(t)$. We have $h_1(t) - h_2(t) = (s(u_x, u_y) - s(v_x, v_y))f$ and because of the condition (5), we only have to show that $\deg(s(u_x, u_y) - s(v_x, v_y)) \leq l$. Suppose that the degree of one of the two polynomials, say $s(u_x, u_y)$, is greater than l. This can only happen if there exist a monomial in $s(x, y) \in R$, say $s_0(x, y) = gx^\alpha y^\beta$, $g \in R$, such that $\deg s_0(u_x, u_y) > l$. If this is the case, then use $\alpha \leq \deg_x s$, $\beta \leq \deg_y s$, (1) and (4) to get:

$$l < \deg(gu_x^\alpha u_y^\beta) \leq \deg g + (\deg u_x)\alpha + (\deg u_y)\beta \leq \deg_t s + d(\deg_x s + \deg_y s) < l,$$

which is a contradiction. Thus $f(t)$ is indeed the irreducible factor of $h_1(t) - h_2(t)$ with highest degree.

The most time consuming part of this decryption algorithm is to factor the polynomial in step 2. In our case this can be done efficiently using the algorithm of Cantor-Zassenhaus [**4**]. This is the one of the fastest methods for factoring polynomials over finite fields.

In the description so far we have omitted all the details in choosing the parameters, which concern the security of the cryptosystem and we have listed only the minimal conditions, which have to be imposed to make the decryption possible. However, the algorithm suggested doesn't always produce a valid decryption. It fails precisely if it happens that $h_1 - h_2 = 0$, i.e. if $s(u_x, u_y) = s(v_x, v_y)$. The probability of this happening is negligible with respect to the degree of s, as discussed in [**1**], where some values for the parameters are suggested. We follow those suggestions for our experiments, described below.

2. Breaking the cryptosystem

We describe an attack which efficiently breaks the protocol just described. However, although it reveals the secret message efficiently, it says nothing about the secret key, as we will see. Finding two or even one rational point on the curve $X(x, y) = 0$ is a hard problem, which it turns out one doesn't need to solve in order to reveal the secret message m. The idea is to work in an extension of R, in which we can find points on the curve and then use these points to evaluate the cypher polynomial.

Let $S = R[y]/(X(x, 0))$ and let $\alpha = \pi(x)$ be the image of x in S under the natural projection

$$\pi : R[x] \to R[x]/(X(x, 0)).$$

The point $(\alpha, 0)$ is on the curve $X(x, y) = 0$, because by construction $X(\alpha, 0) = \pi(X(x, 0)) = 0$. We evaluate the cypher polynomial F at $(\alpha, 0)$ to get

$$F(\alpha, 0) = m + fs(\alpha, 0) + X(\alpha, 0)r(\alpha, 0) = m + fs(\alpha, 0). \tag{7}$$

We now want to go back to our original ring by applying the trace operator. To be precise, recall that we denoted $K = \mathbb{F}_p(t)$ and let $L = S \otimes_R K$. We have the trace operator $Tr : L \to K$, which satisfies $Tr|_K = [L : K]id$.

Now choose an element $0 \neq \beta \in S$ with $Tr(\beta) = 0$. If $\gamma \in S$, but $\gamma \notin R$, then $\beta = \gamma - \frac{Tr(\gamma)}{n}$, where $n = \deg_x X(x, 0)$, is such a choice, provided $(p, n) = 1$ which we assume for simplicity. Indeed,

$$Tr(\gamma - \frac{Tr(\gamma)}{n}) = Tr(\gamma) - Tr(\frac{1}{n})Tr(\gamma) = Tr(\gamma) - Tr(\gamma) = 0,$$

because $Tr(\frac{1}{n}) = \frac{1}{n}[L : K] = \frac{1}{n}\deg(X(0, y)) = 1$.

Now using (7) we get

$$Tr(\beta F(\alpha, 0)) = mTr(\beta) + fTr(\beta s(\alpha, 0)) = fTr(\beta s(\alpha, 0)).$$

In other words, for any choice of β, the adversary can compute $p_\beta = Tr(\beta F(\alpha, 0))$, which is a polynomial in t divisible by f. But f is monic irreducible polynomial of large degree, which allows the adversary to find it, in case $p_\beta \neq 0$. For example he could compute $p_\beta \neq 0$ for several different choices of β, take the greatest common divisor of them, and extract f as the irreducible polynomial of largest degree, which divides the greatest common divisor. The most time consuming computation in this process is the factorization needed to obtain the largest degree irreducible divisor, which is a computation used also in the decryption, as we already saw. We showed how to obtain candidate values for β starting with any $\gamma \in S \setminus R$. The natural choices for γ are α, α^2, $\alpha^3, \ldots$ and in the testing part we never needed to try more than 10 different values for γ.

Now when f is known to the adversary, he can easily obtain m in the following way. Apply Tr operator to equation (7) to get

$$Tr(F(\alpha, 0)) = Tr(m + fs(\alpha, 0)) = nm + nfTr(s(\alpha, 0)). \tag{8}$$

Now nm is the remainder of $F(\alpha, 0)$ when divided by f, because of the conditions $\deg m < l < \deg f$.

The computational steps required for the attack are similar as those required for the decryption, except for the computation of traces. The traces of $\alpha^i, i = 1, \ldots, n-1$ are obtained from the coefficients of $X(x, 0)$ by Newton's identities and, from those values, the trace of any element of S can be easily obtained by linearity. We implemented the encryption, decryption and attack steps in Pari/GP (code available at http://www.ma.utexas.edu/users/voloch/GP/asc.gp) In the key generation part of the testing script we use the following choice of parameters: $w = \deg X$ is a random number between 5 and 8, d is chosen to be 50, the coefficients of $X(x, y) \in R[x, y]$, which are polynomials of t are chosen randomly in such a way that their degree is less than or equal to dw. Finally l is chosen to be a random number between $(2w + 4)d$ and $(2w + 4)d + 100$. Finally we took for p all primes up to 31 and a handful of primes between 32 and 7919. The time it takes to break the system using our attack appears to be a small multiple (depending on p and at most 6 for p in our range) of the decryption time.

3. Variants and other attacks

In addition to the attack described above, the Akiyama-Goto cryptosystem is subject to another attack due to Uchiyama and Tokunaga. Both this attack and ours are discussed in [**2**], where the authors also discuss a new variant of their cryptosystem immune to those attacks. Briefly, this new variant uses the same shape of encryption $F = m+fs+Xr$, except that now $m, f \in \mathbf{F}_p[t, x, y]$. To enable decryption, they need to send two encryptions $F_i = m+fs_i+Xr_i, i = 1, 2$. This new cryptosystem is subject to the following attack:

Let $g = fs_2 - fs_1$. We use a substitution attack as in 6.1.1 of [**2**]. We have $F_2 - F_1 = g + X(r_2 - r_1)$, so by substituting points in a finite field satisfying $X = 0$, we get a set of linear equations for the coefficients of g. So we first find g and then we use a multivariate polynomial factoring algorithm (as in e.g. [**3**]) to find f from g. Once f is found, then we can find m by plugging points satisfying $X = 0$ on $F_i = m + fs_i + Xr_i$ which now are linear in m and s_i.

Another idea of how make the system secure against the attack described in this paper is the following: Go back to the original system and make the cyphertext be of the form $m + fs + Xr + (x^{p^n} - x)b + (y^{p^n} - y)c$ where b and

c are random polynomials in t, x, y. If n is sufficiently large, when we plug in the points U, V we can recover the value of $m + fs$ as the remainder of division by $t^{p^n} - t$, then proceed as before. On the other hand, n cannot be made too large or the system might be subject to a substitution attack. It is not clear whether this new system is efficient or secure and it merits further study.

References

[1] Akiyama, K., Goto, A.: A Public-key Cryptosystem using Algebraic Surfaces (Extended Abstract), PQCrypto Workshop Record, 2006

[2] Akiyama, K., Goto, A.: An improvement of the algebraic surface public-key cryptosystem, Proceedings of SCIS 2008.

[3] Gathen, J. v. z., Kaltofen E., Factorization of multivariate polynomials over finite fields, Math. Comp. 45 (1985), no. 171, 251–261.

[4] Cantor, D., Zassenhaus, H.: A New Algorithm for Factoring Polynomials Over Finite Fields, Mathematics of Computation, 36:587-592, 1981.

Petar Ivanov, Dept. of Mathematics, Univ. of Texas, Austin, TX 78712-0257
E-mail : pivanov@math.utexas.edu

José Felipe Voloch, Dept. of Mathematics, Univ. of Texas, Austin, TX 78712-0257
E-mail : voloch@math.utexas.edu