CONTINUOUS HARD-TO-INVERT FUNCTIONS AND BIOMETRIC AUTHENTICATION

DIMA GRIGORIEV AND SERGEY NIKOLENKO

Abstract. We consider the problem of constructing continuous cryptographic primitives. We present several candidates for continuous hard-to-invert functions. To formulate these candidates, we introduce constructions based on tropical and supertropical circuits.

1. Introduction

Many important cryptographic applications require the underlying primitives to possess some continuity properties. This effect is especially prominent in biometrics: fingerprints, retina scans, and human voices change a little over time, and the conditions are also never exactly the same. However, the system still needs to let the slightly changed human being pass and still needs to deny access for other human beings who have "changed" substantially more. Thus, for biometric applications *continuous* cryptographic primitives would be of great interest.

In biometrics, approaches to continuous cryptography have already been proposed. In [23], a *fuzzy vault scheme* was put forward. In fuzzy vault schemes, continuity is understood in a discrete, set-theoretic sense: a set of features (minutae) is close to another set if their intersection is large and their set difference is small; however, the features themselves remain discrete and must match perfectly, just not all of them. The fuzzy vault scheme of [23] was recently criticized and found vulnerable to certain plausible attacks [38, 39]; however, the general problem of finding continuous primitives remains interesting.

In this work, we propose several candidates for *continuous* hard-to-invert functions, as introduced in [16][1]. We understand continuity in the regular mathematical sense: a continuous function maps close points of a Euclidean space to close points of another Euclidean space.

[1] We deliberately do not use the term "one-way" here, as it has a precise mathematical meaning [13], which we obviously cannot prove for our candidate functions. Hard-to-invert functions are functions which are polynomially easy to compute, but for which there is no known algorithm for inverting them in polynomial time.

This setting makes perfect sense for biometric applications. For example, voice features – spectral and cepstral coefficients and related characteristics – are multidimensional real vectors; a vault signed by somebody's voice should forgive small variations in these integral characteristics.

Our basic cryptographic construction, corresponding to biometric needs, is an authentication scheme based on a one-way function. There exist many secure authentication protocols based on one-way functions, e.g. the Lamport's scheme and the X.509 mechanism based on digital signatures, as well as password schemes [32]. Their exact details are unimportant for our present work; what is important is which function to use as the underlying hard-to-invert function. Suppose that $f : X \to Y$ is a function which is easy to compute but hard to invert, and a participant of the authentication protocol Alice has a secret $x \in X$ (her biometric data). We aim to find a *continuous* function $f : X \to Y$ so that even if Alice's biometrics changed a little over time to some $x' \in X$, the distance $\rho(x, x')$ being small, the value of $f(x')$ would nevertheless be close to $f(x)$, and the other participant of the protocol would be able to authenticate Alice. On the other hand, an impostor Charlie with biometrics $y \in X$ on a relatively large distance $\rho(x, y)$ should not be authenticated, and $f(y)$ should be far from $f(x)$ in Y.

Continuous one-way functions (better to say, function candidates) have already appeared in literature, but examples and discussion have been limited to the field of *physical* one-way functions, i.e. hard-to-invert physical processes and their mathematical models. There are continuous maps based on the second law of thermodynamics that are presumably hard to invert [21]; other physical processes, often naturally continuous, have been proposed as one-way candidates [36, 37]. Our candidates are much simpler, but still do not allow for known efficient inversion algorithms although much thought has been spent on their underlying problems.

For now, no one knows whether there exist functions which are much harder to invert than to compute, especially in the formal cryptographic setting of one-way functions. The hardest results we have without additional assumptions are linear lower bounds (no better explicit lower bounds exist for circuit complexity anyway) [19, 20]. However, we can formulate an open question in theoretical cryptography, which may

This is, obviously, not a mathematical definition, as it relies on our state of knowledge rather than formal concepts; nevertheless, this is the best we can hope for at present.

(or may not, one never knows for sure) turn out to be easier than overcoming these foundational obstacles.

Open question. Provided that one-way functions exist, does there exist a continuous one-way function?

The paper is organized as follows. In Section 2, we consider a polynomial mapping as a one-way candidate. In Section 3, we propose a candidate tropical construction. In Section 4, we give constructions of interactive protocols based on our candidate functions.

2. Polynomial candidates

2.1. **The general idea.** Our first candidate is a polynomial mapping $f : R^n \to R^m$ for $m > n$ (for example, $m = n + 1$) for some ring R. In theory, we usually take $R = \mathbb{R}$ or $R = \mathbb{C}$ and assume that f has integer coefficients; in practice, the corresponding real or rational numbers will all be rational; we denote the rational points of $\mathbb{C}$ by $\mathbb{Q}_{\mathbb{C}} = \mathbb{Q} + i\mathbb{Q}$. Inverting this one-way function is equivalent to solving a (slightly) overdetermined system of polynomial equations:

$$\begin{array}{lcl} f_1(x_1, \dots, x_n) & = & y_1, \\ f_2(x_2, \dots, x_n) & = & y_2, \\ \dots & & \dots \\ f_m(x_1, \dots, x_n) & = & y_m. \end{array}$$

Solving systems of polynomial equations has naturally attracted much attention in algebraic geometry. It is well known that in the worst case, solving even a system of multivariate quadratic equations is NP-complete, both over a finite field in the Turing machine model and in the Blum-Shub-Smale model over an arbitrary ring or field, including $\mathbb{R}$ and $\mathbb{C}$ [2]. It is known that, over a finite field, if m is much larger than n (the system is very much overdetermined), there are efficient heuristics for solving such systems based on linearization, namely the XSL method which was used in a much acclaimed method of breaking block ciphers [6, 8].

Efficient algorithms for solving overdetermined systems on average (which are more relevant in the cryptographic setting) are not known to date. For systems of n homogeneous polynomial equations in exactly $n + 1$ variables (functions $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$), Shub and Smale have developed an ingenious path following (homotopy) method that finds one of their non-zero solutions in average subexponential time [40–44].

However, this method suffers from several restrictions. First, its running time is subexponential in the dimension N of the vector space $\mathcal{H}_{(d)}$ of all homogeneous polynomial maps $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$ with $f = (f_1, \dots, f_n)$, $\deg f_i = d_i$; latest progress in this area brings the average complexity down to $N^{O(\log\log N)}$ [3]. This dimension is polynomial in the number of variables if the degree is constant, and polynomial in the degree if the number of variables is constant, but not both. Second, the path following method is restricted to maps $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$. For overdetermined systems $f : \mathbb{C}^n \to \mathbb{C}^m$, $m > n$, the method does not work, and we have to fall back on Newton's method [10], which only finds a root if one begins in a small enough disc around a zero; see an estimate of the disc radius in [10, Theorem 1].

In order to obtain a small representation of a polynomial with large N, we define a polynomial mapping by an arithmetic circuit. In essence, we allow parentheses in the definition of the polynomial and do not require the system to open them. Arithmetic circuits are able to "conceal" the number of monomials N, specifying a polynomial with an exponential number of monomials by a polynomial size circuit. Even polynomials of exponential degree can sometimes be computed in polynomial time, e.g. the value of $(x+y)^{2^n}$ is easy to compute by repeated squaring (in the Blum-Shub-Smale model, in the bit model the result may have exponential length). Note, however, that many natural questions about circuits in this representation become computationally hard. For example, in [29] it is shown that deciding whether a given polynomial is zero is hard for $\mathrm{P}^{\#\mathrm{P}}$.

2.2. **Continuity modulus.** To use a continuous hard-to-invert function, we also have to specify an estimate on the continuity modulus

$$\omega(f, \delta) = \sup_{|u-v|<\delta} |f(u) - f(v)|,$$

where δ is the maximum distance from the exact stored "password" that should still admit legitimate authentication. For a polynomial, the continuity modulus is bounded only if we restrict our attention to a compact set. Fortunately, in all practical applications there is a compact set $\Omega \subseteq X$ on which it is meaningful to consider the function f (e.g., the set of all reasonably possible fingerprint minutae or mel-frequency cepstral coefficients), so in what follows we assume that all inputs will come from a compact domain Ω.

There are different approaches for computing the continuity modulus. It is easy to compute the continuity modulus of a polynomial. However, we do not know the polynomial's coefficients, we only know its circuit (and remember, computing coefficients may be very hard). We list several ideas.

(1) For a compact set $\Omega \subseteq X$, we can estimate the continuity modulus inductively. For input variables (resp, constants) the

continuity modulus is 1 (resp., 0). For a summation gate, $w_{f+g} \leq w_f + w_g$, so we get a new upper bound by summing the incoming upper bounds. For a multiplication gate,

$$w_{fg} \leq w_f \sup_{x\in\Omega} g(x) + w_g \sup_{x\in\Omega} f(x).$$

The supremum can also be estimated inductively in the obvious way:

$$\sup(f+g) \leq \sup f + \sup g, \quad \sup(fg) \leq \sup f \sup g.$$

However, this estimate loses precision very fast as the size of the circuit grows, so in certain cases it can result in an unacceptably forgiving system.

(2) For a specific $x \in \Omega$, exceedingly imprecise supremum estimates are not necessary, as the continuity modulus reduces to the derivative at point x which can be computed recursively in the obvious way:

$$(f+g)'(x) = f'(x) + g'(x), \quad (fg)'(x) = f'(x)g(x) + f(x)g'(x).$$

(3) The continuity modulus can be estimated even better with the ideas of *interval analysis* developed by Moore [34], Hansen [18], and Matiyasevich [30]; see also recent surveys of the subject [7, 11]. Application of interval analysis to this particular case, especially in the (super)tropical case, may warrant a separate study, so we do not go into details here and note it as an interesting open problem.

In what follows we do not choose a specific approach to computing the continuity modulus but assume that some approach is chosen, and w_f is computed at each node and propagated through the entire circuit.

2.3. **Random key generation.** In order to randomly generate a specific hard-to-invert function candidate, we have to generate a random directed acyclic graph with at least n vertices of indegree zero (inputs and field constants), m vertices of outdegree zero (outputs), and internal vertices with indegree 2 labeled by either "+" or "×". There exist generation models for random directed acyclic graphs, both uniform [31] and based on the ordered graphs model [26]. For use with our protocols, we prefer the latter model for random ordered graph generation, especially given that arithmetic circuits are naturally random ordered graphs. However, in order to keep the output polynomial-sized and to control the continuity modulus we want to produce polynomials of degree $n^{O(1)}$, and random circuits, as noted above, may have exponential degree.

Therefore, we modify the generation model of [26] in order to control the degree. Fix a number n of inputs, a number m of outputs, a degree upper bound D, a number of constant inputs c (all constant inputs in the circuit equal either 1 or -1, and larger constants should be generated from them), and an upper bound on the outdegree $K \geq 2$. The indegree of each non-input vertex is 2 (all gates represent either $+$ or $\times$), and the outdegree is generated randomly when the node is generated. We build a random circuit node by node. Each node is labeled by a pair (s, d), where s is one of x_i, $+$, or $\times$, and d is a natural number representing the "formal degree" of this node. The generation proceeds as follows.

(1) Generate the graph (G, E) with $n + c$ vertices – n with labels $(x_i, 1)$ and c with labels $(\pm 1, 0)$ (the sign is chosen uniformly) – and no edges. Choose outdegrees k_i uniformly from $1..K$ for each vertex and initialize k_i "stubs" for each potential outgoing edge (see [26] for a detailed discussion of these "stubs").

(2) Until m outputs are generated:

  (a) Add a new node x, $G := G \cup \{x\}$, select its label uniformly from $\{+, \times\}$, select two parents y and z uniformly from the "stubs" available at previous vertices, add the corresponding edges $E := E \cup \{(y, x), (z, x)\}$, and delete one "stub" from y and z each.

  (b) Compute the formal degree $\mathrm{fdeg}(x)$:

$$\mathrm{fdeg}(x) = \begin{cases} \max\{\mathrm{fdeg}(y), \mathrm{fdeg}(z)\}, & \text{if } x \text{ is a } +\text{-vertex}, \\ \mathrm{fdeg}(y) + \mathrm{fdeg}(z), & \text{if } x \text{ is a } \times\text{-vertex}. \end{cases}$$

  (c) Compute the continuity modulus w_x (see 2.2).

  (d) If $\mathrm{fdeg}(x) \geq \lfloor \frac{D}{2} \rfloor + 1$, mark x as an output and do not generate outgoing "stubs" for it. Otherwise, generate k outgoing "stubs", where k is chosen uniformly from $1..K$.

(3) Delete remaining "stubs" and output (G, E).

Obviously, this generation model will, with overwhelming probability, generate m outputs of formal degree from $\frac{D}{2}$ to D in time polynomial in $n + m + c$.

Note that the "formal degree" $\mathrm{fdeg}(x)$ is merely an upper bound on the actual degrees of the generated polynomials; the actual degree may be much lower due to cancellations. However, each output will have a degree $\deg f_i \leq D$; the worst case is a product of two gates of degree $\lfloor \frac{D}{2} \rfloor$.

In what follows, we assume that there exists a polynomial-time procedure $\mathrm{Gen}(n, m, D)$ that produces an arithmetic circuit for a polynomial map $f : \mathbb{Q}_{\mathbb{C}}^n \to \mathbb{Q}_{\mathbb{C}}^m$ of degrees $\deg f_i \leq D$.

2.4. **The resulting protocol.** To make a precise example, let us specify a simple secret-key authentication protocol. Suppose that agent A (Alice) wants to authenticate with a server S using her biometric data. At the beginning of the protocol, S stores the biometric data x, and Alice possesses her data x', presumably close to x. The algorithm parameters include n (dimension) and ϵ (authentication precision).

(1) A initiates the protocol and represents her biometric data as a vector $x' \in \mathbb{Q}_{\mathbb{C}}^n$.
(2) S randomly selects an arithmetic circuit f with n input variables as shown in 2.3 and sends a representation of this circuit to A.
(3) A randomly selects a vector $r \in \mathbb{Q}_{\mathbb{C}}^n$ and a scalar $\alpha \in \mathbb{Q}_{\mathbb{C}}$ (this is analogous to random padding), computes $f(r + \alpha x')$ and transmits (r, α, y) for $y = f(r + \alpha x')$.
(4) S computes ω, the continuity modulus at point $r + \alpha x$, with any method from Section 2.2 and checks that $||y - f(r + \alpha x)|| \leq \omega\epsilon$. If so, S accepts the authentication of A.

A passive adversary in this protocol is faced with the problem of solving a system of polynomial equations $f(r + \alpha x) = a$ with respect to the unknown x for f specified as an arithmetic circuit. If a passive adversary has observed k runs of this protocol for the same server and agent, he faces a problem of solving a system

$$f^1(r^1 + \alpha^1 x) = a^1, f^2(r^2 + \alpha^2 x) = a^2, \ldots, f^k(r^k + \alpha^k x) = a^k.$$

Note that it is hard for an adversary to apply the methods of [6, 8] because the monomials of the polynomial f are unknown, and there are a lot of them.

It would be desirable for the server S to store only images of f, i.e., y rather than x; this would reduce the danger of identity theft. However, in this simple protocol it is near impossible

3. Supertropical candidates

Exact algebraic approaches to solving a system of nonlinear equations include the resultant approach and Gröbner bases. Computing the resultant, despite recent advances [5], is impractical for large multivariate cases [24, 25]. Gröbner bases provide a more practical framework [12, 25], but still, the complexity of exact (symbolic) methods for solving large systems of polynomial equations is too large.

However, in our case it would suffice to find an approximate solution; for approximate solutions, there are also Newton's method (the homotopy continuation method is based on it), and optimization approaches (see, e.g., [35] that combines several of these methods). To spoil Newton's method, it is enough, in theory, to make the number of equations not equal to the number of variables, and we have done so in the previous section. However, to avoid both Newton's method in practice and optimization approaches, one would like the system's function f to have many local minima; it would be even better if the function had many kinks and/or breaks so that there would be no gradient to follow or it would be misleading.

Both of these properties come together in *tropical* constructions [4, 22, 33]; moreover, counterparts of symbolic methods are not known for the tropical case. Tropical algebras are based on the *tropical semiring* (also known as the min-plus algebra) which is a subset of reals with an infinity point closed under addition, with two operations:

$$x \oplus y = \min(x, y), \quad x \otimes y = x + y.$$

A tropical monomial $m = a \otimes x_{i_1} \otimes \ldots \otimes x_{i_n} = a + x_{i_1} + \ldots + x_{i_n}$, $1 \leq i_j \leq n$, is simply a linear function, while a tropical polynomial $p = m_1 \oplus \ldots \oplus m_k = \min(m_1, \ldots, m_k)$ is a minimum of several linear functions, i.e., a concave piecewise linear function with several discontinuity regions.

Several cryptographic constructions based on tropical algebras have been recently presented in [17]. For the purposes of continuous cryptographic constructions, however, we would like to extend the tropical semiring by one more operation, namely regular multiplication (we do not introduce a special symbol for it and use $\cdot$ and juxtaposition). We call the resulting extended semiring $(A, \cdot, \otimes, \oplus)$, $A \subseteq \mathbb{R} \cup \{\infty\}$, a *supertropical algebra*. In the supertropical algebra, a supertropical monomial is in fact a polynomial

$$m(x_1, \ldots, x_n) = x_1^{i_{11}} x_2^{i_{12}} \ldots x_n^{i_{1n}} \otimes \ldots \otimes x_1^{i_{m1}} x_2^{i_{m2}} \ldots x_n^{i_{mn}},$$

and a supertropical polynomial

$$p(x_1, \ldots, x_n) = m_1(x_1, \ldots, x_n) \oplus \ldots \oplus m_k(x_1, \ldots, x_n)$$

is a minimum of several polynomial functions, i.e., a piecewise polynomial function which is not necessarily concave anymore and still has a lot of discontinuity regions.

We represent a supertropical polynomial system of n variables with a directed acyclic graph with at least n vertices of indegree zero (inputs and field constants), m vertices of outdegree zero (outputs), and internal vertices with indegree 2 labeled by either "$\cdot$", "$\oplus$", or "$\otimes$". The generation protocol remains the same with the following additions: for an $\oplus$-gate x with parents y and z that compute functions f and g, respectively,

$$\begin{aligned} \mathrm{fdeg}(x) &= \max\{\mathrm{fdeg}(y), \mathrm{fdeg}(z)\}, \\ w_{f\oplus g} &= \max\{w_f, w_g\}; \end{aligned}$$

the multiplication and $\otimes$-gates (i.e., the usual addition) are treated in the same way as their polynomial counterparts. The protocol from Section 2.4 works in a similar fashion. The continuity modulus computation and random key generation are done similar to the algorithms presented in subsections 2.2 and 2.3, respectively.

As for the hardness of the resulting protocol, little is known, but there are reasons to believe that it is hard to solve systems of polynomial tropical equations. In [17], it has been shown that it is NP-hard to find a solution of a system of tropical polynomial equations. This does not mean that there are no algorithms efficient on average, in the generic case, or for our particular choice of key generation, but it is usually an indicator that this is indeed a hard problem. For example, only very recently D. Grigoriev has presented an algorithm for solving a system of *linear* tropical equations [14]; this problem is known to be in NP∩co-NP, and it is suspected to have polynomial complexity, but Grigoriev's algorithm has been recently shown to require superpolynomial time in the worst case [9]. In an independent result, Akian, Gaubert, and Guterman presented several weakly polynomial algorithms for this problem [1]. Many important invariants of tropical systems (varieties) are hard to compute [27, 28, 45]. As for the supertropical case, the outlook is even bleaker; we do not know of any works in this direction, but it is obvious that solving a supertropical system is at least as hard as solving a tropical system and at least as hard as solving a polynomial system. Based on all of the above, we recommend our supertropical candidate for use with the protocol of Section 2.4.

4. An interactive protocol

In this section, we describe one more class of protocols that can be implemented in a continuous fashion with polynomial and supertropical circuits. The basic protocol relies upon the hardness of the matrix conjugation problem. The protocol has been presented in [15]; it is an interactive authentication scheme that goes, for an underlying matrix ring G, as follows.

(1) Alice's public key is a pair of matrices $(A, X^{-1}AX)$, where $A \in G$, $X \in G$; Alice's secret key is the matrix X.
(2) For his challenge, Bob selects a random matrix $B \in G$ and a random non-invertible endomorphism φ of the ring G. Bob sends B and φ to Alice.
(3) Alice responds with random positive integers p and q and asks Bob to send back random nonzero constants c_1, c_2, and c_3 so that the new (better randomized) challenge is $B' = c_1A + c_2B + c_3A^pB^q$.
(4) Alice responds with $\varphi(X^{-1}B'X)$.
(5) Bob selects a random word $w(x, y)$ (without negative exponents), evaluates

$$M_1 = w\left(\varphi(A), \varphi(B')\right), \qquad M_2 = w\left(\varphi(X^{-1}AX), \varphi(X^{-1}B'X)\right),$$

and computes their traces. If $tr(M_1)$ is sufficiently close to $tr(M_2)$, Bob accepts authentication, otherwise he rejects.

In [15], Grigoriev and Shpilrain propose to use the ring of $n \times n$ matrices over sparse truncated k-variate polynomials over a finite field (in [15], $\mathbb{Z}_{11}$ is suggested). We propose to use the key generation process of Section 2.3 to generate matrices over k-variate polynomials over an infinite field $\mathbb{F}$. Note that for an infinite field itself, there would be another way for the adversary: compute the private key X from the public key (A, C), find the space of solutions for the equation $AX = XC$ and sample a matrix X' at random; with probability 1, X' will be nondegenerate. Thus, the protocol in this section would be insecure for infinite fields. For polynomial rings, this is not a problem because a matrix can be invertible only if the determinant of the matrix is a constant polynomial (has degree zero), an event of probability zero. Note also that this protocol does not work over the (super)tropical semiring at all since the only invertible tropical matrices are monomial matrices, i.e., products of a diagonal and a permutation matrix, which would make the break trivial [4].

Each matrix element can be represented as an arithmetic circuit; matrix products involve a linear number of additions and multiplications and can be implemented without significantly increasing the circuit size. The following remarks should be made about this process.

(1) In the protocol, the matrices A and B do not have to be invertible, so no problems arise with its generation. The matrix X, however, has to be invertible, and therefore we propose to generate it as a product of elementary matrices, i.e., matrices

that have exactly one non-zero element. The non-zero element is generated as in Section 2.3.

(2) To generate a random endomorphism, one can generate $\varphi : x_i \to f_i$, where f_i are random truncated k-variate polynomials over $\mathbb{F}$ with zero constant term.

(3) To define what "sufficiently close" means on step 5 of the protocol, Bob uses the continuity moduli for each element of M_1 and M_2 and computes $\omega_{tr(M_1)}$ and $\omega_{tr(M_2)}$ as the continuity moduli of the corresponding sums of elements.

To break this protocol, an adversary would have to solve a system of n^2 polynomial equations given as arithmetic circuits. Note that it would be hard even for an infinite field because the size of a linearization grows exponentially and, as shown in [15], even for a reasonable choice of protocol parameters the linear system becomes too large to solve.

5. Conclusion

In this paper, we have presented two hard-to-invert candidates that share a common desirable property: they are *continuous*. When preserved in an authentication protocol, this property allows for small changes in the secret information so that a sufficiently close authentication request (e.g., slightly modified biometrics) is still accepted. We have also introduced supertropical algebras as a platform for cryptographic protocols.

We have presented the ideas of two protocols. Further work about these protocols should deal with their specific implementations and tuning the parameters in order to test their properties and modify them to be as secure and efficient in practice as possible.

Acknowledgements. This work has resulted from the cooperation organized under the Federal Target Programme "Scientific and scientific-pedagogical personnel of the innovative Russia". The first author is grateful to the Max-Planck Institut für Mathematik, Bonn for its hospitality during writing this paper. Work of the second author has been supported by the Russian Presidential Grant Programme for Young Ph.D.'s, grant no. MK-6628.2012.1, for Leading Scientific Schools, grant no. NSh-3229.2012.1, and Russian Fund for Basic Research grants 11-01-12135-ofi-m-2011 and 11-01-00760-a.

References

[1] Marianne Akian, Stephane Gaubert, and Alexander Guterman. Tropical polyhedra are equivalent to mean payoff games. *International Journal on Algebra and Computations*, To appear, 2011.

[2] Lenore Blum, Michael Shub, and Stephen Smale. On a theory of computation and complexity over the real numbers: Np-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, 1989.

[3] Peter Bürgisser and Felipe Cucker. Solving polynomial equations in smoothed polynomial time and a near solution to Smale's 17th problem. In *Proceedings of the* 42^{nd} *Annual ACM Symposium on Theory of Computing*, pages 503–512, 2010.

[4] P. Butkovic. *Max-linear systems: theory and algorithms*. Springer-Verlag London, 2010.

[5] Arthur D. Chtcherba. *A new Sylvester-type resultant method based on the Dixon-Bezout formulation*. PhD thesis, The University of New Mexico, 2003.

[6] Carlos Cid and Gaëtan Leurent. An analysis of the XSL algorithm. In *Proceedings of the* 11^{th} *ASIACRYPT, International Conference on the Theory and Application of Cryptology and Information Security, Lecture Notes in Computer Science*, volume 3788, pages 333–352, 2005.

[7] Michael J. Cloud, Ramon E. Moore, and R. Baker Kearfott. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[8] Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *Proceedings of the* 8^{th} *ASIACRYPT, International Conference on the Theory and Application of Cryptology and Information Security, Lecture Notes in Computer Science*, volume 2501, pages 323–337, 2002.

[9] Alexey Davydow. Complexity bounds on Grigoriev's algorithm for solving tropical linear systems. To appear.

[10] J. P. Dedieu and Michael Shub. Newton's method for overdetermined systems of equations. *Mathematics of Computation*, 69(231):1099–1115, 1999.

[11] Oliver Didrit, Luc Jaulin, and Michel Kieffer. *Applied Interval Analysis*. Springer, Berlin, 2001.

[12] Ralf Fröberg. *An Introduction to Gröbner Bases*. Wiley & Sons, 1997.

[13] Oded Goldreich. *Foundations of Cryptography. Basic Tools*. Cambridge University Press, 2001.

[14] Dima Grigoriev. Complexity of solving tropical linear systems. MPI Preprint 2010-60, to appear in Computational Complexity, 2010.

[15] Dima Grigoriev and Vladimir Shpilrain. Authentication from matrix conjugation. *Groups, Complexity, and Cryptology*, 1:199–205, 2009.

[16] Dima Grigoriev and Vladimir Shpilrain. Zero-knowledge authentication schemes from actions on graphs, groups, or rings. *Annals of Pure and Applied Logic*, 162:194–200, 2010.

[17] Dima Grigoriev and Vladimir Shpilrain. Tropical cryptography. Preprint MPI 2011-11, 2011.

[18] E. R. Hansen. A generalized interval arithmetic. In *Interval Mathematics, Lecture Notes in Computer Science*, volume 29, pages 7–18. Springer-Verlag, Berlin, 1978.

[19] Alain P. Hiltgen. Constructions of feebly-one-way families of permutations. In *Proc. of AsiaCrypt '92*, pages 422–434, 1992.

[20] Edward A. Hirsch and Sergey I. Nikolenko. A feebly secure trapdoor function. In *Proceedings of the* 4^{th} *Computer Science Symposium in Russia, Lecture Notes in Computer Science*, volume 5675, pages 129–142, 2009.

[21] Norbert Hungerbühler and Michael Struwe. A one-way function from thermodynamics and applications to cryptography. *Elemente der Mathematik*, 58(2):2026–2030, 2003.

[22] I. Itenberg, Grigory Mikhalkin, and E. Shustin. *Tropical algebraic geometry*, volume 35 of *Oberwolfach Seminars*. Birkhäuser, Basel, 2007.

[23] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.

[24] D. Kapur and Y. N. Lakshman. Elimination methods: an introduction. In B. Donald, D. Kapur, and J. Mundy, editors, *Symbolic and Numerical Computation for Artificial Intelligence*. Academic Press, 1992.

[25] D. Kapur and T. Saxena. Comparison of various multivariate resultant formulations. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, pages 187–195. ACM Press, 1995.

[26] Brian Karrer and M. E. J. Newman. Random acyclic networks. *Physical Review Letters*, 102(12):128701, 2009.

[27] K. H. Kim and F. W. Roush. Factorization of polynomials in one variable over the tropical semiring. ArXiv preprint math/0501167, 2005.

[28] K. H. Kim and F. W. Roush. Kapranov rank vs. tropical rank. *Proceedings of the American Mathematical Society*, 134:2487–2494, 2006.

[29] Pascal Koiran and Sylvain Perifel. The complexity of two problems on arithmetic circuits. *Theoretical Computer Science*, 389(1–2):172–181, 2007.

[30] Yuri V. Matiyasevich. A posteriori interval analysis. In *Proceedings of the EUROCAL'85, Lecture Notes in Computer Science*, volume 204, pages 328–334, 1985.

[31] G. Melançon, I. Dutour, and M. Bousquet-Mélou. Random generation of directed acyclic graphs. *Electronic Notes in Discrete Mathematics*, 10:202–207, 2001.

[32] Alfred Menezes, Paul van Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography*. Boca Raton, Florida: CRC Press, 1997.

[33] Grigory Mikhalkin. Tropical geometry and its applications. In *Proceedings of the International Congress of Mathematicians*, volume 2, pages 827–852, 2006.

[34] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliff, New Jersey, 1966.

[35] Victor Y. Pan and Ai-Long Zheng. New progress in real and complex polynomial root-finding. *Computers & Mathematics with Applications*, 61(5):1305–1334, 2011.

[36] Srinivasa Ravikanth Pappu. *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, 2001.

[37] Srinivasa Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[38] Hoi Ting Poon and Ali Miri. A collusion attack on the fuzzy vault scheme. *The ISC International Journal of Information Security*, 1(1):27–34, 2009.

[39] Walter J. Schreier and Terrance E. Boult. Cracking fuzzy vaults and biometric encryption. In *Proceedings of the Biometrics Symposium 2007 at The Biometric Consortium Conference (BCC), Baltimore, Maryland, USA*, 2007.

[40] Michael Shub and Stephen Smale. Complexity of Bezout's theorem I: Geometric aspects. *Journal of the American Mathematical Society*, 6:459–501, 1993.

[41] Michael Shub and Stephen Smale. Complexity of Bezout's theorem II: Volumes and probabilities. In F. Eyssette and A. Galligo, editors, *Computational Algebraic Geometry*, volume 109 of *Progress in Mathematics*, pages 267–285. Bikrhäuser, 1993.

[42] Michael Shub and Stephen Smale. Complexity of Bezout's theorem III: Condition number and packing. *Journal of Complexity*, 9:4–14, 1993.

[43] Michael Shub and Stephen Smale. Complexity of Bezout's theorem V: Polynomial time. *Theoretical Computer Science*, 134:141–164, 1994.

[44] Michael Shub and Stephen Smale. Complexity of Bezout's theorem IV: Probability of success; extensions. *SIAM Journal of Numerical Analysis*, 33:128–148, 1996.

[45] T. Theobald. On the frontiers of polynomial computations in tropical geometry. *Journal of Symbolic Computation*, 41:1360–1375, 2006.

CNRS, Mathématiques, Université de Lille, France
E-mail address: dmitry.grigoryev@math.univ-lille1.fr

Steklov Mathematical Institute, St. Petersburg, Russia; St. Petersburg Academic University, St. Petersburg, Russia
E-mail address: sergey@logic.pdmi.ras.ru