Continuous hard-to-invert functions

D. Yu. Grigoriev S. I. Nikolenko

April 5, 2012

Outline

Motivation

- Many important cryptographic applications require the underlying primitives to possess some continuity properties.
- Biometrics: fingerprints, retina scans, and human voices change a little over time, and the conditions are also never exactly the same.
- For biometric applications, *continuous* cryptographic primitives would be of great interest.

Fuzzy vault scheme

- [Juels, Sudan, 2006]: fuzzy vault scheme – a discrete version of continuity.
- A set of features (minutae) is close to another set if their intersection is large and their set difference is small.
- The protocol was further advanced and implemented, but then...

Fuzzy vault scheme

- [Juels, Sudan, 2006]: fuzzy vault scheme – a discrete version of continuity.
- A set of features (minutae) is close to another set if their intersection is large and their set difference is small.
- The protocol was further advanced and implemented, but then...
- [Schreier, Boult, 2007]: "Cracking Fuzzy Vaults...".
- [Poon, Miri, 2009] – another attack.
- We propose an idea for cryptographic primitives continuous in the common sense of the word.

Polynomial systems

- Candidate 1: a polynomial mapping $f : R^n \to R^m$ for $m > n$ (for example, $m = n + 1$) for some ring R (we usually take $R = \mathbb{R}$ or $R = \mathbb{C}$ and assume that f has integer coefficients).
- Inverting f is equivalent to solving a (slightly) overdetermined system of polynomial equations:

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

Polynomial systems

- How do we solve polynomial equations?
- Worst-case: NP-hard already for quadratic equations (finite field, rational numbers, algebraic numbers, Turing machine or Blum-Shub-Smale model in an arbitrary field).
- Over a finite field, if m is much larger than n, we can linearize: XSL method.
- Systems of n homogeneous equations in $n+1$ variables: Shub-Smale homotopy method with average-case complexity $N^{O(\log\log N)}$, where N is the dimension of the space of all such homogeneous polynomial maps $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$ [Bürgisser, Cucker, 2010].

Polynomial systems

- For overdetermined systems we basically only have Newton's method and variations.
- Newton's method has to start in a small enough neighborhood around the zero in question; there are estimates on the size of the neighborhood [Dedieu, Smale, 1999].
- To make a polynomial system hard, we need to:
  - make N large (increase the degree and dimension of the system);
  - consider a system with many local minima to make Newton's method fail.

Arithmetic circuits

- To increase degree, we specify polynomials with arithmetic circuits.
- Some polynomials of very large degree have compact circuit representations.
- Example: $(x+y)^{2^n}$ has a small circuit representation.
- Many natural questions about circuits in this representation become computationally hard.
- E.g., deciding whether a given polynomial is zero is hard for $\mathrm{P}^{\#\mathrm{P}}$ [Koiran, Perifel, 2007].

Continuity modulus

- To use a continuous hard-to-invert function, we 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.
- Consider a polynomial defined over a compact domain Ω (of meaningful values of f).

Continuity modulus

- We can get an upper bound by induction on the circuit size:
  1. for input variables (resp, constants) the continuity modulus is 1 (resp., 0);
  2. 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;
  3. for a multiplication gate,

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

where the supremum can also be estimated inductively:

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

Continuity modulus

- However, this bound becomes less and less exact as the size of the circuit grows, and in certain cases it can result in an unacceptably forgiving system.
- But for a specific $x \in \Omega$, we can estimate the continuity modulus as the derivative at point x which can be computed recursively:

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

Protocol

- A simple authentication protocol. Alice (A) 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.
  1. A initiates the protocol and represents her biometric data as a vector $x' \in \mathbb{C}^n$.
  2. S randomly selects an arithmetic circuit f with n input variables and sends a representation of this circuit to A.
  3. A randomly selects a vector $r \in \mathbb{C}^n$ and a scalar $\alpha \in \mathbb{C}$ (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 $r + \alpha x$, and checks that $\|y - f(r + \alpha x)\| \le \omega\epsilon$. If so, S accepts the authentication of A.
- How do we "randomly select a circuit"?

Key generation

- Circuits are directed graphs.
- 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.

Key generation

1. Generate the graph (G, E) with $n + c$ vertices, n with labels $(x_i, 1)$ and c with labels $(\pm 1, 0)$.
2. Choose outdegrees k_i uniformly from $1..K$ for each vertex and initialize k_i "stubs" for each potential outgoing edge.
3. Until m outputs are generated:
   1. Add a new node x, $G := G \cup \{x\}$, select its label, 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.
   2. Compute the formal degree $\mathrm{fdeg}(x)$:
   $$\mathrm{fdeg}(x) = \begin{cases} \max\{\mathrm{fdeg}(y), \mathrm{fdeg}(z)\}, & \text{if } x \text{ is a } +\text{-vertex}, \\ \mathrm{fdeg}(y) + \mathrm{fdeg}(z), & \text{if } x \text{ is a } \times\text{-vertex}. \end{cases}$$
   3. Compute the continuity modulus w_x.
   4. 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 $1..K$.
4. Delete remaining "stubs" and output (G, E).

Protocol

- A simple authentication protocol. Alice (A) 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.
  1. A initiates the protocol and represents her biometric data as a vector $x' \in \mathbb{C}^n$.
  2. S randomly selects an arithmetic circuit f with n input variables and sends a representation of this circuit to A.
  3. A randomly selects a vector $r \in \mathbb{C}^n$ and a scalar $\alpha \in \mathbb{C}$ (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 $r + \alpha x$, and checks that $\|y - f(r + \alpha x)\| \leq \omega\epsilon$. If so, S accepts the authentication of A.
- What does an adversary have to do?

Protocol

- A passive adversary in this protocol has to solve 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

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

- Note that it is hard for an adversary to linearize because the monomials of f^i are numerous and, even better, unknown.

Outline

How do we spoil Newton's method?

- We said that, to defeat Newton's method, we'd like to make sure f has a lot of local minima, so gradient descent has low probability of success.
- But it would be even better if f had no gradient at all! (or, at least, was not differentiable often enough)
- That's why we consider *tropical* constructions.

Tropical and supertropical circuits

- 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:

$$m = a \otimes x_{i_1} \otimes \ldots \otimes x_{i_n} = a + x_{i_1} + \ldots + x_{i_n},\ 1 \leq i_j \leq n.$$

- A tropical polynomial

$$p = m_1 \oplus \ldots \oplus m_k = \min(m_1, \ldots, m_k)$$

is a concave piecewise linear function with several discontinuity regions.

Tropical and supertropical circuits

- For our continuous cryptographic constructions, we extend the tropical semiring by regular multiplication and call the resulting extended semiring $(A, \cdot, \otimes, \oplus)$, $A \subseteq \mathbb{R} \cup \{\infty\}$, a *supertropical algebra*.
- A supertropical monomial is in fact a polynomial

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

- 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.

Tropical and supertropical circuits

- The protocol remains the same, only circuits are now supertropical (fdeg and continuity modulus for a $\oplus$-gate are just max of its parents).
- As for the underlying problem, much less is known than for regular polynomials, but these are hard problems.
- There is currently no polynomial algorithm for solving even *linear* tropical systems; only very recently weakly polynomial algorithms appeared [Grigoriev 2010; Akian, Gaubert, Guterman, 2011].
- Tropical polynomial systems are obviously NP-hard; there are no known good algorithms.
- For supertropical linear and polynomial systems, nothing is known (except that they are obviously at least as hard as tropical ones).

An interactive protocol

- We suggest a candidate interactive protocol, too, following [Grigoriev, Shpilrain, 2009].
- It relies upon the hardness of matrix conjugation.

An interactive protocol

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 $B' = c_1 A + c_2 B + c_3 A^p B^q$.
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.

An interactive protocol

- We propose to use this protocol for multivariate polynomials over an infinite field $\mathbb{F}$.
- Note that for an infinite field itself, the adversary could 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.
- But for polynomial rings, a random matrix is invertible with probability zero (its determinant must have degree zero).
- Unfortunately, over the (super)tropical semiring the protocol does not work at all: the only invertible tropical matrices are monomial [Butkovic, 10].

Thank you!

Thank you for your attention!