On a family of collision-free functions

by Attila Bérczes, János Folláth and Attila Pethő
University of Debrecen

CECC'09, Trebič, June 23, 2009

1. On one-way functions

Important building blocks for:
- many cryptographic protocols,
- verifying passwords,
- creating digital signatures,
- constructing secure pseudo-random-number generators.

Informally a one-way function is a function which is "easy" to compute but "hard" to invert.

Complexity theoretical: a one-way function can be computed in low degree polynomial time, but all of its inverses only in exponential time.

Important properties of one-way functions are
- collision resistance
- avalanche effect.

Goldwasser and Bellare (2001): Let S be a finite set. A function $f : S \mapsto S$ is called **collision resistant** if there exists a constant $c \geq 0$, such that

$$\mathbf{P}\left[f(x_1) = f(x_2)\right] \leq |S|^{-c}$$

holds, where (x_1, x_2) is considered as a uniformly distributed vector variable over S^2.

There is another property of f to find x_2 for fixed input x_1 or to find x_2 for fixed output $f(x_1)$ such that $f(x_1) = f(x_2)$. These properties are called **2nd-preimage resistance** and **preimage resistance** respectively.

Bérczes, Ködmön and Pethő (2004) constructed a family of collision resistant functions based on norm functions, well studied in the theory of diophantine equations.

Bérczes and Járási (2008) extended this family with index forms. In both cases the functions were reduced modulo m, where m is the product of two large primes. For security reason m should have at least 1024 binary digits.

The first construction was implemented by the company Crypto Ltd under the name CODEFISH.

J.-P. Aumasson (2009) pointed out some vulnerability of the implemented algorithm.

The aim of this work is to continue the above investigations and improve their results in two directions:

• we are working on finite fields, thus the length of the output can be shorter, e.g. 256 or 512 bits.

• we can handle functions over finite fields of characteristic two, which can make the implementation of the proposed algorithms much more efficient.

The main difference of the new construction is that our function is the sum of a homogenous polynomial of degree $n > 1$ and a linear polynomial.

2. Main results

Let p be a prime and let $q = p^f$ with $f \geq 1$ an integer. Denote by $\mathbb{F}_q$ the finite field with q elements.

Theorem 1. *Let* $f(\mathbf{X}) \in \mathbb{F}_q[X_1, \ldots, X_m]$ *be a polynomial*

$$f(\mathbf{X}) := b(X_1, \ldots, X_m) + a(X_1, \ldots, X_m)$$

with homogeneous polynomials $a(\mathbf{X}), b(\mathbf{X})$ *satisfying* $k = \deg a(\mathbf{X}) < \deg b(\mathbf{X}) = n$, $\deg_{X_i} b(\mathbf{X}) = n$ *for* $1 \le i \le n$. *Suppose that there exist* $1 \le j_1 < j_2 \le n$ *such that*

$$b_0(X_{j_1}, X_{j_2}) := b(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0) \qquad (1)$$

has no multiple zero.

Let $N(f, \gamma, q)$ *denote the number of solutions of the equation* $f(x_1, \ldots, x_m) = \gamma$ *in* $x_1, \ldots, x_m \in \mathbb{F}_q$. *Then*

$$|N(f, \gamma, q) - q^{m-1}| \le (n-1)(n-2)q^{m-3/2} + 5n^{13/3}q^{m-2}. \qquad (2)$$

Moreover, if $q > 15n^{13/3}$, *then*

$$|N(f, \gamma, q) - q^{m-1}| \le (n-1)(n-2)q^{m-3/2} + (5n^2+n+1)q^{m-2}. \qquad (3)$$

As a simple corollary we get that the functions defined in Theorem 1 are collision resistant.

Corollary 1. *Assume that the polynomial* $f(\mathbf{X}) \in \mathbb{F}_q[X_1, \ldots, X_m]$ *satisfies the requirements of Theorem 1. Denote* $P_{coll}(f, \gamma)$ *the probability that* $f(\mathbf{x})$ *assumes the value* $\gamma \in \mathbb{F}_q^*$*, when* $\mathbf{x}$ *runs through uniformly on the elements* $\mathbb{F}_q^m$*. Then*

$$P_{coll}(f, \gamma) \leq \frac{1}{q} + \frac{(n-1)(n-2)}{q^{3/2}} + \frac{5n^{13/3}}{q^2}.$$

Moreover, if $q > 5n^{13/3}$*, then*

$$P_{coll}(f, \gamma) \leq \frac{3}{q}.$$

3. Auxiliary results

Estimations of the number of $\mathbb{F}_q$ points lying on a hypersurface defined over $\mathbb{F}_q$ have long history: Hasse (1933), Lang and Weil (1954), W.M. Schmidt (1974). The best result is due to Cafure and Matera (2006).

Theorem 2. *For an absolutely irreducible $\mathbb{F}_q$ -hypersurface H of $\mathbb{A}^n$ of degree δ the following estimate holds:*

$$\left||H \cap \mathbb{F}_q^n| - q^{n-1}\right| \leq (\delta-1)(\delta-2)q^{n-3/2} + 5\delta^{13/3}q^{n-2}.$$

If q is large enough, much better remaining term was proved by Cafure and Matera (2006).

Theorem 3. *Let $q > 15\delta^{13/3}$ and let $H \subseteq \mathbb{A}^n$ be an absolutely irreducible $\mathbb{F}_q$ - hypersurface of degree δ. Then the following estimate holds:*

$$\left||H \cap \mathbb{F}_q^n| - q^{n-1}\right| \leq (\delta-1)(\delta-2)q^{n-3/2} + (5\delta^2+\delta+1)q^{n-2}.$$

The next lemma shows that under certain condition lacunary polynomials are absolutely irreducible.

Lemma 1. *Let K be any field, and fix an algebraic closure $\overline{K}$ of K. Let $n \geq 4$ be an integer and let $G(X,Y) = Y^n + A(X)Y^{n-1} + B(X) \in K[X,Y]$ be a polynomial with the properties $A(X), B(X) \in K[X]$, $B(X)$ has no multiple zeros and $\deg A(X) \neq \deg B(X) \geq 1$. Then $G(X,Y)$ is irreducible over $\overline{K}$, i.e. it is absolutely irreducible.*

The proof of this lemma is similar as the proof of the Schöneman-Eisenstein irreducibility proof.

Lemma 2. *Let K be any field. Let $f(\mathbf{X}) \in K[X_1, \ldots, X_m]$ be a polynomial such that*

$$f(\mathbf{X}) := b(X_1, \ldots, X_m) + a(X_1, \ldots, X_m)$$

with homogeneous polynomials $a(\mathbf{X}), b(\mathbf{X})$ satisfying $k = \deg a(\mathbf{X}) < \deg b(\mathbf{X}) = n$, $\deg_{X_i} b(\mathbf{X}) = n$ for $1 \leq i \leq n$. Further, suppose that there exist indices $1 \leq j_1 < j_2 \leq n$ such that the binary form

$$b_0(X_{j_1}, X_{j_2}) := b(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0) \qquad (4)$$

has no multiple zero. Then the polynomial $f(\mathbf{X}) + \gamma$ is absolutely irreducible for every $0 \neq \gamma \in K$.

4. Proof of Theorem 1 and its Corollary

Proof of Theorem 1 It follows from Lemma 2 that the polynomial $f - \gamma$ is absolutely irreducible over $\mathbb{F}_q$.

Thus by Theorems 2 and 3 the result follows.□

Proof of the Corollary Obviously, $\mathbb{F}_q^m$ has q^m elements and $P_{coll}(f, \gamma) = \frac{N(f,\gamma,q}{|\mathbb{F}_q^m|}$, which together with Theorem 1 implies the first statement immediately.

If $q > 5n^{13/3}$ then $q^{1/2} > (n-1)(n-2)$ and we get the second statement from the first one at once. □

5. Practical considerations

There are two typical ways for the choice of the finite field; either q is a prime, or q is a power of 2.
To avoid brute force attack the binary length of q must be at least 128.
The computation time depends very much on m, we decided to choose $m = 4$.

We decided to choose $f(\mathbf{X}) := b(\mathbf{X}) + a(\mathbf{X})$ such that $b(\mathbf{X})$ and $a(\mathbf{X})$ are in diagonal form, i.e. $b(\mathbf{X}) = \beta_1 X_1^r + \ldots + \beta_m X_m^r$ and $a(\mathbf{X}) = \alpha_1 X_1^s + \ldots \alpha_m X_m^s$ with $0 < s < r < q$ and $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m \neq 0$. With this choice all assumptions of Theorem 1 automatically hold except that the polynomial $b_0(X_i, X_j) = \beta_i X_i^r + \beta_j X_j^r$ has no multiple roots.

The polynomial $b_0(X_i, X_j)$ has multiple roots in $\bar{\mathbb{F}}_q$ if and only if $c(\mathbf{X}) = X^r + \gamma$ with $\mathbf{X} = \mathbf{X}_i/X_j$ and $\gamma = \beta_j/\beta_i$ has multiple roots in $\bar{\mathbb{F}}_q$.

It is well-known that the multiple roots of $c(\mathbf{X})$ are roots of $\gcd(c(\mathbf{X}), c'(\mathbf{X}))$. Since $c'(\mathbf{X}) = r\mathbf{X}^{r-1}$, it is non-zero if r and the characteristic of $\mathbb{F}_q$ are coprime. This holds for all r, if q is a prime, and for all odd r, if $q = 2^f$. Further, if $c'(\mathbf{X}) \neq 0$, then its only root is 0, which is a zero of $c(\mathbf{X})$ if and only if $\gamma = 0$, but this is excluded by the choice of the β's. Thus we proved the following assertion.

Proposition 1. *Let* $f(\mathbf{X}) := b(\mathbf{X}) + a(\mathbf{X})$ *such that* $b(\mathbf{X}) = \beta_1 X_1^r + \ldots + \beta_m X_m^r$, $a(\mathbf{X}) = \alpha_1 X_1^s + \ldots + \alpha_m X_m^s$ *and* $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m \neq 0$. *If* $0 < s < r < q$ *and* r *is odd if* $q = 2^f$, *then* $f(\mathbf{X})$ *satisfies all assumptions of Theorem 1.*

The coefficients of $b(\mathbf{X})$ and $a(\mathbf{X})$ should be chosen distinct random elements of $\mathbb{F}_q$. Further, if we choose r such that $q > 5r^{13/3}$, then by Corollary 1 the probability that $f(\mathbf{X})$ takes a fixed element of $\mathbb{F}_q$ is at most $1/q$. If q is a prime, then we put $s = 1$, i.e. $a(\mathbf{X})$ a linear polynomial.

If $q = 2^f$, then choose a normal basis of $\mathbb{F}_q$ and represent the elements in this basis. Then squaring means a periodic left shift, while multiplication with a fixed element means mixing of the co-ordinates. Thus a good choice of r is, if its binary representation has at least 7 non-zero digits. Since the highest and the lowest digits are one, the remaining five ones should be distributed among the remaining positions.

To be more specific, let $f = 128$. Then, for example $r = 2^{28} + 2^{24} + 2^{20} + 2^{15} + 2^{10} + 2^5 + 1 = 286295073$ satisfies all requirements. We propose to choose the exponent $s < r$ on the same principles as r.

We implemented $f(\mathbf{X})$ with several choice of the parameters. The result of the computational time is displayed in the following table.

Hash length	Characteristic	Kilobyte/second
256	odd	338
512	odd	121
254	even	8
509	even	6

6. Remarks on the implementation

By the implementation of the proposed functions choosing the field has a great impact on the performance. Beyond the obvious importance of the field size the choice of the characteristic has the greatest significance. It depends on the characteristic whether we can use simple modular arithmetic (which is advantageous on general purpose processors) or the (with hardware fast implementable) even characteristic arithmetic can be applied.

Prime field arithmetic The (odd) prime field arithmetic (which means simple modular arithmetic in practice) is better suited for general purpose processors. On general purpose processors a single difficulty arises in conjunction with the proposed algorithm: the size of the operands.

There exists many implementation of arbitrary precision arithmetic for various programming languages. They are well tested and optimized, accelerated by assembly language fragments. Some of them also takes a staged approach to the multiplication and squaring algorithms. Still, arbitrary precision arithmetic is significantly slower than word-level arithmetic. Consequently the performance of the proposed hash algorithm will be only comparable to those algorithms that also have to use arbitrary precision arithmetic . We used the GNU Multiprecision Library in our implementation.

Even characteristic arithmetic The primary strength of the proposed construction lies in the hardware implementation. If we define the function over an even characteristic field and use a normal basis representation, the squaring can be done with a simple cyclic shift which is extremely fast. The normal basis multiplication is also well studied and multiple fast architectures and implementations were proposed Agnew et al. (1991,1993). Consequently the even characteristic version is a viable practical hash function when a hardware solution is needed.

The implementations of even characteristic multiplication on general purpose processors are usually slower than the prime field arithmetic, but since the fast normal basis squaring, it may be also worth of consideration. Normal basis multiplication algorithms require many bit level operations, and thats why the implementations cannot make use of the full data path of the processor. The algorithms proposed by Reyhani-Masoleh and Hasan

(199?) avoid this disadvantage, making the multiplication much faster. This algorithm can only be applied by fields having a type even gaussian normal base (GNB), which exists for $GF(2^k)$ only if k is not divisible by 8. GNB type t exists if and only if $p = tk+1$ is prime and gcd$(\frac{tk}{l}, k) = 1$), where l is the multiplicative order of 2 modulo p. By the tests we used GNB type 2 in both case. Our implementation is only a pure c reference implementation of the construction, it can be significantly accelerated by further optimization,