Generalized Iterated Hash Functions Revisited: New Complexity Bounds for Multicollision Attacks

Tuomas Kortelainen [1] Juha Kortelainen[2] Ari Vesanen [2]

[1]Mathematics Division, Department of Electrical and Information Engineering, University of Oulu

[2]Department of Information Processing Science, University of Oulu

Indocrypt 2012, December 8 – 12, 2012, Kolkata, India

Outline

Hash Function

Assume (without loss of generality) that all our messages are nonempty words over the (message block) alphabet A. Let $B = \{0, 1\}$ be the binary alphabet.

Definition

A *hash function* (of length n, where $n \in \mathbb{N}_+$) is a mapping $g : A^+ \to B^n$.

Here B^n gives us all possible hash values.

Hash Function

Assume (without loss of generality) that all our messages are nonempty words over the (message block) alphabet A. Let $B = \{0, 1\}$ be the binary alphabet.

Definition

A *hash function* (of length n, where $n \in \mathbb{N}_+$) is a mapping $g : A^+ \to B^n$.

Here B^n gives us all possible hash values.

Multicollisions

Let $k \in \mathbb{N}_+$. A *k-collision* in the hash function g is a set $C \subseteq A^+$ such that $|C| = k$ and $g(x) = g(y)$ for all $x, y \in C$.

K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota (2008) [6]: A k-collision in g can be found (with probability approx. $\frac{1}{2}$) by hashing $(k!)^{\frac{1}{k}} 2^{\frac{n(k-1)}{k}}$ messages. This is known as (generalized) *birthday paradox*.

Two remarks can be made immediately:

Multicollisions

Let $k \in \mathbb{N}_+$. A *k-collision* in the hash function g is a set $C \subseteq A^+$ such that $|C| = k$ and $g(x) = g(y)$ for all $x, y \in C$.

K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota (2008) [6]: A k-collision in g can be found (with probability approx. $\frac{1}{2}$) by hashing $(k!)^{\frac{1}{k}} 2^{\frac{n(k-1)}{k}}$ messages. This is known as (generalized) *birthday paradox*.

Two remarks can be made immediately:

Multicollisions

Let $k \in \mathbb{N}_+$. A *k-collision* in the hash function g is a set $C \subseteq A^+$ such that $|C| = k$ and $g(x) = g(y)$ for all $x, y \in C$.

K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota (2008) [6]: A k-collision in g can be found (with probability approx. $\frac{1}{2}$) by hashing $(k!)^{\frac{1}{k}} 2^{\frac{n(k-1)}{k}}$ messages. This is known as (generalized) *birthday paradox*.

Two remarks can be made immediately:

Multicollisions (2)

- In the case $k = 2$ approximately $\sqrt{2} \cdot 2^{\frac{n}{2}}$ hashings are needed; intuitively most of us would expect the number to be around 2^{n-1}.
- For each k in $\mathbb{N}_+$, finding a $(k+1)$-collision consumes much more resources than finding a k-collision.

Multicollisions (2)

- In the case $k = 2$ approximately $\sqrt{2} \cdot 2^{\frac{n}{2}}$ hashings are needed; intuitively most of us would expect the number to be around 2^{n-1}.
- For each k in $\mathbb{N}_+$, finding a $(k+1)$-collision consumes much more resources than finding a k-collision.

Compression Function

Definition

A *compression function* (of length n) is a mapping $f : B^n \times A \to B^n$ where A is an alphabet, $n \in \mathbb{N}_+$, and $|A| > 2^n$.

An ideal compression function f is a *fixed input length random oracle* (FIL-RO for short): for each $h \in B^n$ and $y \in A$, the value $f(h, y) \in B^n$ is chosen uniformly at random.

Compression Function

Definition

A *compression function* (of length n) is a mapping $f : B^n \times A \to B^n$ where A is an alphabet, $n \in \mathbb{N}_+$, and $|A| > 2^n$.

An ideal compression function f is a *fixed input length random oracle* (FIL-RO for short): for each $h \in B^n$ and $y \in A$, the value $f(h, y) \in B^n$ is chosen uniformly at random.

Iterated hash function

Let now f be a compression function; define the *iterated hash function* $f^+ : B^n \times A^+ \to B^n$ (based on f) inductively as follows. Let $h \in B^n$, $y_1 \in A$, and $y_2 \in A^+$. Then $f^+(h, y_1) = f(h, y_1)$ and $f^+(h, y_1 y_2) = f^+(f(h, y_1), y_2)$.

We assume that the attacker knows compression function f only as a black box. She/he does not know anything about the internal structure of f, but can make *queries* on f and get the respective *responses*. The attacker tries to achieve her/his goal while minimizing the number of these queries.

Iterated hash function

Let now f be a compression function; define the *iterated hash function* $f^+ : B^n \times A^+ \to B^n$ (based on f) inductively as follows. Let $h \in B^n$, $y_1 \in A$, and $y_2 \in A^+$. Then $f^+(h, y_1) = f(h, y_1)$ and $f^+(h, y_1 y_2) = f^+(f(h, y_1), y_2)$.

We assume that the attacker knows compression function f only as a black box. She/he does not know anything about the internal structure of f, but can make *queries* on f and get the respective *responses*. The attacker tries to achieve her/his goal while minimizing the number of these queries.

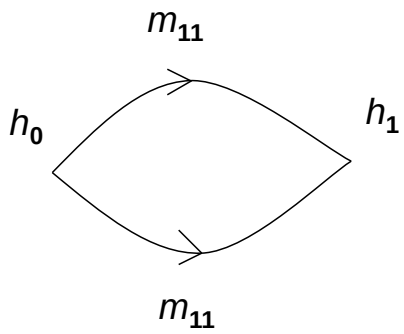
Multicollision on Iterated Hash Function

Definition

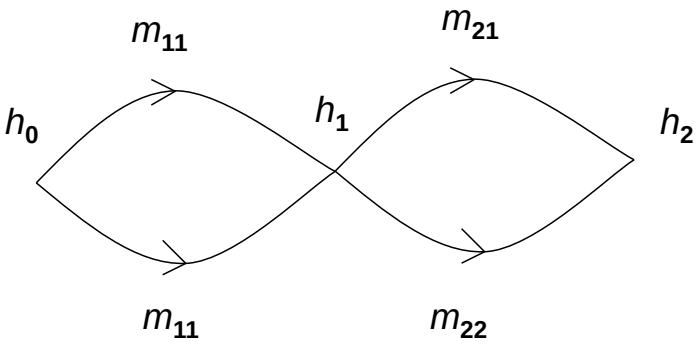
Given $k \in \mathbb{N}_+$ and $h_0 \in B^n$, a *k-collision* (*with initial value* h_0) *in the iterated hash function* f^+ is a set $C \subseteq A^+$ such that $|C| = k$ and for all $u, v \in C$, $|u| = |v|$ and $f^+(h_0, u) = f^+(h_0, v)$.

Joux's attack

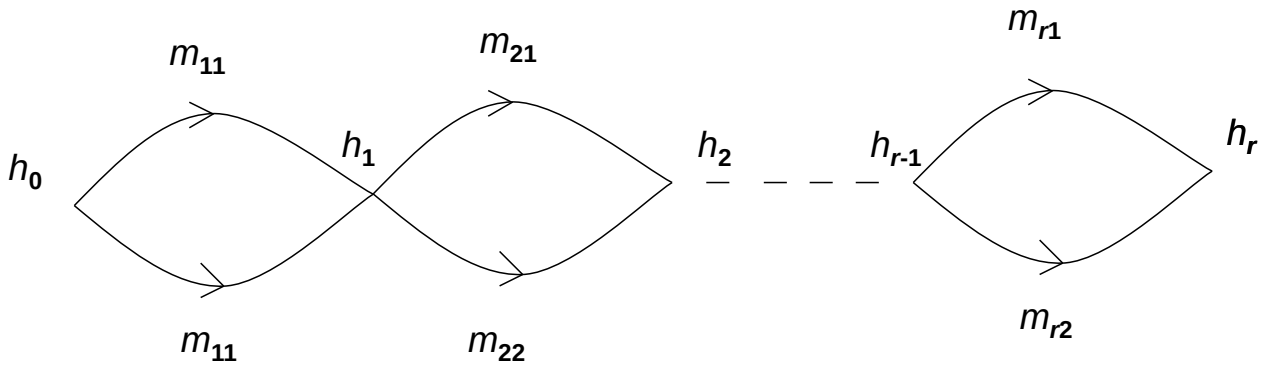
The Joux's method [1] for generating 2^k-collisions is of the (average) $O(k \cdot 2^{\frac{n}{2}})$ compression function calls. In this method, one can begin with any initial value h_0 and one finds two message blocks x_1 and x_1' such that $h_1 = f(h_0, x_1) = f(h_0, x_1')$. Now, continuing the same approach to h_1 and so on, one obtains a 2^k-collision after finding k two-collisions.



$f(h_0,m_{11})= f(h_0,m_{12})=h_1$



$f(h_0,m_{11})= f(h_0,m_{12})=h_1$ $\quad$ $f(h_1,m_{21})= f(h_1,m_{22})=h_2$



$f(h_0,m_{11})= f(h_0,m_{12})=h_1$ $f(h_1,m_{21})= f(h_1,m_{22})=h_2$ $f(h_{r-1},m_{r1})= f(h_{r-1},m_{r2})=h_r$

Generalized iterated hash functions

Let $l \in \mathbb{N}_+$ and $\alpha = i_1 i_2 \cdots i_s$, where $i_j \in \{1, 2, \cdots l\}$ for all $j = 1, 2, \cdots s$. We define *generalized iterated hash function* (based on α and f) by setting

$$f_\alpha(h_0, b_1 b_2 \cdots b_l) = f^+(h_0, b_{i_1} b_{i_2} \cdots b_{i_s}).$$

The question arises whether or not the idea of the Joux attack can be applied in this more broad setting, i.e., can Joux's approach be used in the multicollision in certain generalized iterated hash functions?

Generalized iterated hash functions

Let $l \in \mathbb{N}_+$ and $\alpha = i_1 i_2 \cdots i_s$, where $i_j \in \{1, 2, \cdots l\}$ for all $j = 1, 2, \cdots s$. We define *generalized iterated hash function* (based on α and f) by setting

$$f_\alpha(h_0, b_1 b_2 \cdots b_l) = f^+(h_0, b_{i_1} b_{i_2} \cdots b_{i_s}).$$

The question arises whether or not the idea of the Joux attack can be applied in this more broad setting, i.e., can Joux's approach be used in the multicollision in certain generalized iterated hash functions?

Example of a generalized iterated hash function

Assume that we have message $x = x_1x_2x_3$ and $\alpha = 1 \cdot 2 \cdot 3 \cdot 2 \cdot 1 \cdot 3$.

Now $f_\alpha(h_0, x) = f^+(h_0, x_1x_2x_3x_2x_1x_3)$. This means that the Joux attack can not be directly carried out.

Example of a generalized iterated hash function

Assume that we have message $x = x_1 x_2 x_3$ and $\alpha = 1 \cdot 2 \cdot 3 \cdot 2 \cdot 1 \cdot 3$.

Now $f_\alpha(h_0, x) = f^+(h_0, x_1 x_2 x_3 x_2 x_1 x_3)$. This means that the Joux attack can not be directly carried out.

Word combinatorics

It turns out that it is possible to apply the idea of the Joux under certain assumptions. This happens by using the regularities in word α.

In combinatorics of words, the theory of 'unavoidable regularities' usually concerns properties of long words over a fixed finite alphabet (Ramsay, Shirshov and van der Waerde). The approach induced by generalized iterated hash functions is a bit different.

We study unavoidable regularities which are satisfied by any sufficiently long words in which the number of occurences of each symbol is bounded by a fixed constant.

Word combinatorics

It turns out that it is possible to apply the idea of the Joux under certain assumptions. This happens by using the regularities in word α.

In combinatorics of words, the theory of 'unavoidable regularities' usually concerns properties of long words over a fixed finite alphabet (Ramsay, Shirshov and van der Waerde). The approach induced by generalized iterated hash functions is a bit different.

We study unavoidable regularities which are satisfied by any sufficiently long words in which the number of occurences of each symbol is bounded by a fixed constant.

Word combinatorics

It turns out that it is possible to apply the idea of the Joux under certain assumptions. This happens by using the regularities in word α.

In combinatorics of words, the theory of 'unavoidable regularities' usually concerns properties of long words over a fixed finite alphabet (Ramsay, Shirshov and van der Waerde). The approach induced by generalized iterated hash functions is a bit different.

We study unavoidable regularities which are satisfied by any sufficiently long words in which the number of occurences of each symbol is bounded by a fixed constant.

Generalized 2-bounded hash functions

In their article Nandi and Stinsson [5] assume that each message block can by used only once or twice. In another words a single symbol can appear in α only once or twice. They were able to show that under these assumpitions in order to create a 2^k-collision the number of compression function calls the attacker needs is

$$O(k^2 lnk(n + lnln2k)2^{\frac{n}{2}}).$$

Generalized q-bounded hash functions

Hoch and Shamir chose in their article [2] even broader setting studying generalized $q-$bounded hash functions. This means that a single message block can be used (or appear in α) at most q times. Results of this article were further studied in [4]. The main result is, that the expected value of creating 2^k-collision is at most

$$2.5 \cdot q \cdot 2^{2^{2q-3}} k^{(2q-3)2^{2q-1}} n^{(q-1)^2 2^{2q-1}} 2^{\frac{n}{2}}$$

compression function calls.

More advanced method

The article [3] used word combinatorics to create more efficient algorithm for attacking $q-$bounded hash functions. This meant that it was possible to show that the expected value of compression calls needed to create 2^k-collision was at most

$$2.5 \cdot q \cdot k^{(2q-3)2^q} n^{(q-1)^2 2^q} 2^{\frac{n}{2}}.$$

About our article

The approach on $q-$ bounded generalized hash functions has been based on both word combinatorics and classical famous algebraic and combinatorial results (such as Dilworth's Theorem and Hall's Matching Theorem). In this work, we put aside these classical results and look at the problem purely as it is: a question of combinatorics on finite strings of symbols.

We introduce a set of new concepts and tools to search unavoidable regularities in long q-restricted words. Our results allow us to create multicollision attacks on generalized iterated hash functions with substantially smaller amount of resources than has been known before.

About our article

The approach on $q-$ bounded generalized hash functions has been based on both word combinatorics and classical famous algebraic and combinatorial results (such as Dilworth's Theorem and Hall's Matching Theorem). In this work, we put aside these classical results and look at the problem purely as it is: a question of combinatorics on finite strings of symbols.

We introduce a set of new concepts and tools to search unavoidable regularities in long q-restricted words. Our results allow us to create multicollision attacks on generalized iterated hash functions with substantially smaller amount of resources than has been known before.

Our main result

Theorem

Let m, n and q be positive integers such that $m > n$ and $q \geq 2$ and $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ a compression function. Then, for each $k \in \mathbb{N}_+$, there exists a 2^k-collision attack on the generalized iterated hash function f_α such that the expected number of queries on f is at most

$$2,5 \cdot q \cdot 5^{q-2} \cdot 2^{\lceil \log_2 n \rceil \frac{(q+4)q(q-1)}{6} + \lceil \log_2 k \rceil q} \cdot 2^{\frac{n}{2}}$$

.

What this would mean in practise

Assume that $n = 256$, $k = 4$ and $q = 3$. Expected number of compression function calls required are aproximately

Articles [2, 4]: $7.5 \cdot 2^{8630}$

Article [3]: $7.5 \cdot 2^{432}$

This study: $7.5 \cdot 5 \cdot 2^{190}$. This is well below $\sqrt[16]{16!} \cdot 2^{240}$ offered by the brute force attack (see article [6]).

What this would mean in practise

Assume that $n = 256$, $k = 4$ and $q = 3$. Expected number of compression function calls required are aproximately

Articles [2, 4]: $7.5 \cdot 2^{8630}$

Article [3]: $7.5 \cdot 2^{432}$

This study: $7.5 \cdot 5 \cdot 2^{190}$. This is well below $\sqrt[16]{16!} \cdot 2^{240}$ offered by the brute force attack (see article [6]).

What this would mean in practise

Assume that $n = 256$, $k = 4$ and $q = 3$. Expected number of compression function calls required are aproximately

Articles [2, 4]: $7.5 \cdot 2^{8630}$

Article [3]: $7.5 \cdot 2^{432}$

This study: $7.5 \cdot 5 \cdot 2^{190}$. This is well below $\sqrt[16]{16!} \cdot 2^{240}$ offered by the brute force attack (see article [6]).

What this would mean in practise

Assume that $n = 256$, $k = 4$ and $q = 3$. Expected number of compression function calls required are aproximately

Articles [2, 4]: $7.5 \cdot 2^{8630}$

Article [3]: $7.5 \cdot 2^{432}$

This study: $7.5 \cdot 5 \cdot 2^{190}$. This is well below $\sqrt[16]{16!} \cdot 2^{240}$ offered by the brute force attack (see article [6]).

References I

- Joux, A. Multicollisions in iterated hash functions. Aplication to cascaded constructions. In Franklin, M.K., ed: Advances in Cryptology - CRYPTO '04. In LNCS 3152 (2004) 306-316
- Hoch, J., Shamir, A. Breaking the ICE - finding multicollisions in iterated concatenated and expanded (ICE) hash functions. In LNCS 4047 (2006) 179-194.
- Kortelainen, J., Kortelainen, T., Vesanen, A. Unavoidable regularities in long words with bounded number of symbol occurrences. In LNCS 6842 (2011) 519-530.
- Kortelainen, J., Halunen, K., Kortelainen, T. Multicollision Attacks and Generalized Iterated Hash Functions. In JMC 4 (2010) 239-270.

References II


Nandi, M., Stinson, D.R.: Multicollision attacks on some generalized sequential hash functions. IEE Transactions on Information Theory 53(2) (2007) 759-767


Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K.: Birthday paradox for multicollisions. IEICE Transactions 91-A(1)(2008) 39-45