Differentiability of four prefix-free PGV hash functions

Kuwakado, Hidenori

Hirose, Shoichi

(Citation)
IEICE Electronics Express,6(13):955-958

(Issue Date)
2009

(Resource Type)
journal article

(Version)
Version of Record

IEICE Electronics Express, Vol.6, No.13, 955–958

Differentiability of four prefix-free PGV hash functions

Hidenori Kuwakado[1a)] **and Shoichi Hirose**[2]

[1] *Graduate School of Engineering, Kobe University, 1–1 Rokkodai-cho Nada-ku Kobe, 657–8501, Japan*

[2] *Graduate School of Engineering, The University of Fukui, 3–9–1 Bunkyo Fukui City, 910–8507, Japan*

a) *kuwakado@kobe-u.ac.jp*

Abstract: A Preneel-Govaerts-Vandewalle hash function with a prefix-free padding method (PF-PGV hash function) is a block-cipher-based iterative hash function. Chang *et al.* claimed that 16 PF-PGV hash functions were indifferentiable from a random oracle. This letter shows that 4 out of 16 PF-PGV hash functions are not indifferentiable from a random oracle.

Keywords: information security, hash function, indifferentiability, random oracle

Classification: Science and engineering for electronics

References

[1] J. Black, P. Rogaway, and T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from PGV," *Advances in Cryptology - CRYPTO 2002, Lecture Notes in Computer Science*, vol. 2442, pp. 320–335, 2002.

[2] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya, "Merkle-Damgård revisited: How to construct a hash function," *Advances in Cryptology - CRYPTO 2005, Lecture Notes in Computer Science*, vol. 3621, pp. 430–448, 2005.

[3] D. Chang, S. Lee, M. Nandi, and M. Yung, "Indifferentiable security analysis of popular hash functions with prefix-free padding," *Advances in Cryptology - ASIACRYPT 2006, Lecture Notes in Computer Science*, vol. 4284, pp. 283–298, 2006.

[4] H. Kuwakado and S. Hirose, "Formalizing security of hash functions against length-extension attacks," *IEICE Technical Report*, vol. 108, no. 474, pp. 273–276, 2009.

1 Introduction

Applications of cryptographic hash functions include authentication systems, public-key schemes, digital signature schemes, and multi-party protocols. Since the security of these cryptosystems has been often proved in the ran-

DOI: 10.1587/elex.6.955
Received May 27, 2009
Accepted June 06, 2009
Published July 10, 2009

dom oracle model, constructing a hash function that is indifferentiable from a random oracle has been attracting interest.

A Preneel-Govaerts-Vandewalle hash function with a prefix-free padding method (PF-PGV hash function) is an iterative hash function based on a block cipher. A PGV hash function has 64 variations for connection and constant addition [1]. Coron *et al.* [2] have shown that the Davies-Meyer hash function with a prefix-free padding method, which is one of 64 PF-PGV hash functions, is indifferentiable from a random oracle. After then, Chang *et al.* [3] have claimed that 16 PF-PGV hash functions are indifferentiable from a random oracle. This letter shows that 4 out of the 16 PF-PGV hash functions are not indifferentiable from a random oracle[1].

2 Definition and known results

Let $E(k,x)$ be an ideal cipher from $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}^n$ and let $D(k,y)$ be its decryption where k is a key, x is a plaintext, and y is a ciphertext. A hash function H from $\{0,1\}^*$ to $\{0,1\}^n$ is constructed with E. Consider an adversary A that distinguishes between H and a random oracle R from $\{0,1\}^*$ to $\{0,1\}^n$. Then, the advantage of A is defined as

$$\mathrm{Adv}_H(A) = \Pr\left[A^{H^E,E,D} = 1\right] - \Pr\left[A^{R,S^R,T^R} = 1\right],$$

where S and T are simulators of E and D, respectively [2, 3]. Notice that simulators S, T do not see the adversary's queries to R. A hash function H is said to be *indifferentiable* from the random oracle R if there are simulators S, T such that $\mathrm{Adv}_H(A)$ is negligible for any efficient adversary A. Conversely, H is said to be *differentiable* from the random oracle R if there is an adversary A such that $\mathrm{Adv}_H(A)$ is not negligible for any efficient simulators S, T.

Given a message m, a prefix-free padding method Pad(), and a compression function F derived from E, a PF-PGV hash function H is defined as follows:

1. Split the padded message $\mathrm{Pad}(m)$ into t n-bit message blocks $m_1, m_2, \ldots, m_t$.
2. Let h_0 be a public constant. For $i = 1$ to t, compute $h_i = F(h_{i-1}, m_i)$.
3. Output h_t as $H(m)$.

Two examples of prefix-free padding methods are given below [2].

Prefix-free padding method 1: The padding method consists in prepending the message size in bits as the first message block. The last message block is then padded with bit '1' followed by zeros. This padding method assumes that the message size is less than 2^n.

Prefix-free padding method 2: The padding wastes one bit per a message block where a message block includes $n-1$ bits of a message. The one bit is bit '1' if the message block is the last one, otherwise it is bit '0'.

[1]The preliminary version of this letter appeared in [4].

DOI: 10.1587/elex.6.955
Received May 27, 2009
Accepted June 06, 2009
Published July 10, 2009

Table I. PGV hash functions. ($w_i = h_{i-1} \oplus m_i$, v: constant)

No.	$F(h_{i-1}, m_i)$	No.	$F(h_{i-1}, m_i)$
1	$E(m_i, h_{i-1}) \oplus h_{i-1}$	9	$E(w_i, m_i) \oplus v$
2	$E(m_i, w_i) \oplus w_i$	10	$E(w_i, m_i) \oplus w_i$
3	$E(m_i, h_{i-1}) \oplus w_i$	11	$E(m_i, h_{i-1}) \oplus v$
4	$E(m_i, w_i) \oplus h_{i-1}$	12	$E(w_i, h_{i-1}) \oplus v$
5	$E(w_i, m_i) \oplus m_i$	13	$E(m_i, h_{i-1}) \oplus m_i$
6	$E(w_i, h_{i-1}) \oplus h_{i-1}$	14	$E(w_i, h_{i-1}) \oplus w_i$
7	$E(w_i, m_i) \oplus h_{i-1}$	15	$E(m_i, w_i) \oplus v$
8	$E(w_i, h_{i-1}) \oplus m_i$	16	$E(m_i, w_i) \oplus m_i$

Note: The index of functions in [3] is different from that in [1]. The above table follows the index of [3].

The compression function F derived from E is written as

$$F(h_{i-1}, m_i) = E(a, b) \oplus c$$

where $a, b, c \in \{h_{i-1}, m_i, w_i, v\}$, $w_i = h_{i-1} \oplus m_i$, and v is an n-bit public constant. Accordingly, there are 64 PF-PGV hash functions.

Coron *et al.* [2] have proved that the Davies-Meyer PF-PGV hash function (No.1 in Table I) is indifferentiable from a random oracle in the ideal cipher model. Chang *et al.* [3] have claimed that 16 PF-PGV hash functions in Table I were indifferentiable from a random oracle in the ideal cipher model. From results of [1] and [3], other 48 PF-PGV hash functions are differentiable from a random oracle.

3 Distinguishing attack

This section shows that PF-PGV hash functions using the following compression functions F (No. 11, 13, 15, 16 in Table I) are differentiable from a random oracle.

$$F(m_i, h_{i-1}) = E(m_i, h_{i-1}) \oplus v, \quad F(m_i, h_{i-1}) = E(m_i, h_{i-1}) \oplus m_i,$$
$$F(m_i, h_{i-1}) = E(m_i, w_i) \oplus v, \quad F(m_i, h_{i-1}) = E(m_i, w_i) \oplus m_i.$$

It should be noted that four compression functions are invertible if a message block m_i is known. This property allows an adversary to distinguish between these PF-PGV hash functions and a random oracle.

As an example, we show a distinguishing attack against the 11th PGV hash function, $F(m_i, h_{i-1}) = E(m_i, h_{i-1}) \oplus v$, with the padding method 1. Let $(\Phi, \Psi_e, \Psi_d) \in \{(H, E, D), (R, S, T)\}$. Consider an adversary A defined below.

1. Choose an n-bit string $m_1^{(1)}$ and an $n-1$-bit string $m_2^{(1)}$ at random.
2. Let $m^{(1)} = m_1^{(1)} \parallel m_2^{(1)}$ where $\parallel$ denotes the concatenation operator on strings. Make a query $m^{(1)}$ to oracle Φ. The answer $\Phi(m^{(1)})$ is denoted by $h^{(1)}$.

DOI: 10.1587/elex.6.955
Received May 27, 2009
Accepted June 06, 2009
Published July 10, 2009

3. Ask the value of $\Psi_d(m_2^{(1)} \parallel 1, h^{(1)} \oplus v)$ to oracle Ψ_d. The answer is denoted by x.
4. Choose an $n-1$-bit string $m_2^{(2)}$ at random.
5. Ask the value of $\Psi_e(m_2^{(2)} \parallel 1, x)$ to oracle Ψ_e. The answer is denoted by y.
6. Let $m^{(2)} = m_1^{(1)} \parallel m_2^{(2)}$. Make a query $m^{(2)}$ to oracle Φ. The answer $\Phi(m^{(2)})$ is denoted by $h^{(2)}$.
7. If $y \oplus v = h^{(2)}$, output 1, otherwise output 0.

If $(\Phi, \Psi_e, \Psi_d) = (H, E, D)$, then the probability that A outputs 1 is 1. Next, suppose $(\Phi, \Psi_e, \Psi_d) = (R, S, T)$. Since simulators S, T are not allowed to see the communication between A and R (i.e., $m^{(1)}, h^{(1)}, m^{(2)}$ and $h^{(2)}$), the simulators cannot know $m_1^{(1)}$ and $h^{(2)}$. However, the simulators have to answer y satisfying $y \oplus v = h^{(2)}$ to behave like E, D. In other words, the simulators have to guess the value of $h^{(2)} = R(m_1^{(1)} \parallel m_2^{(2)})$ without knowing $m_1^{(1)}$. No simulators can correctly guess $h^{(2)}$ with a probability larger than 2^{-n} because R is the random oracle. Hence, the advantage of A is $\mathrm{Adv}_H(A) = 1 - 2^{-n}$, which means that this PF-PGV hash function H is differentiable from the random oracle R. Similar attacks are applicable to other three PF-PGV hash functions. Even if the prefix-free padding method 2 is used, the above attack can be modified without loss of advantage $\mathrm{Adv}_H(A)$.

4 Conclusion

This letter has demonstrated the distinguishing attack to disprove Chang *et al.*'s claim, that is, their claim is partially incorrect. No attack for distinguishing between the other 12 PF-PGV hash functions and a random oracle has been found.

Acknowledgments

We thank Dr. Yoshida, Dr. Ideguchi, and Dr. Owada at Hitachi, Ltd. and Prof. Ohta and Prof. Sakiyama at The University of Electro-Communications for fruitful discussions. This research was supported by the National Institute of Information and Communications Technology, Japan.

DOI: 10.1587/elex.6.955
Received May 27, 2009
Accepted June 06, 2009
Published July 10, 2009