Instructions for Authors of SBC Conferences Papers and Abstracts

Giovani Ferreira[1], **Rafael Marconi**[1]

[1]CEUB - Centro Universitário de Brasília
Caixa Postal 4488 – 70.904-970 – Brasília – DF – Brazil

Abstract. *adhaskjdhaksjd [Tanenbaum and Van Steen 2002]*

1. Introduction

Apesar de tecnicas para evitar o ataque do aniversario ja terem sido criadas [Aiello and Venkatesan 1996].

Collision search is an important tool in cryptanalysis. A broad range of cryptanalytic problems such as computing discrete logarithms, finding hash function collisions, and meet-in-the-middle attacks can be reduced to the problem of finding two distinct inputs, a and b, to a function f such that $f(a) = f(b)$ [Van Oorschot and Wiener 1999].

The most common cryptographic uses of hash functions are with digital signatures and for data integrity. With digital signatures, a long message is usually hashed (using a publicly available hash function) and only the hash-value is signed. The party receiving the message then hashes the received message, and verifies that the received signature is correct for this hash-value. This saves both time and space compared to signing the message directly, which would typically involve splitting the message into appropriate-sized blocks and signing each block individually. Note here that the inability to find two messages with the same hash-value is a security requirement, since otherwise, the signature on one mes- sage hash-value would be the same as that on another, allowing a signer to sign one message and at a later point in time claim to have signed another [Menezes et al. 1996]. Hash functions may be used for data integrity as follows. The hash-value corresponding to a particular input is computed at some point in time. The integrity of this hash-value is protected in some manner. At a subsequent point in time, to verify that the input data has not been altered, the hash-value is recomputed using the input at hand, and compared for equality with the original hash-value. Specific applications include virus protection and software distribution [Menezes et al. 1996].

2. Related Concepts

2.1. Hash Functions

A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values [Menezes et al. 1996].

2.2. Hash Collision

Collision Resistance - It is computationally infeasible to find any two distinct input x, x' which hash to the same output, i.e., such that $h(x) = h(x')$. (Note that here there is free choice of both inputs.) [Menezes et al. 1996]

A hash function h is called $collisionfree$, if it maps messages of any length to strings of some fixed length, but such that finding x, y with $h(x) = h(y)$ is a hard problem. Note that we are concentrating here on publicly computable hash functions, i.e. functions that are not controlled by a secret key [Damgård 1989].

Hash functions are designed to take a message of arbitrary bitlength and map it to a fixed size output called a hash result. Let $H : M \rightarrow R$ be such a hash function. Typically, hash functions are constructed from a function $h : B \times R \rightarrow R$ which takes a fixed size block of message bits together with an intermediate hash result and produces a new intermediate hash result. A given message $m \in \mathbb{M}$ is typically padded to a multiple of the block size and split into blocks $m_1, m_2, ..., m_l \in B$. The padding often includes a field which indicates the number of bits in the original message. Beginning with some constant $r_0 \in \mathbb{R}$, the sequence $r_i = h(m_i, r_{i-1})$ is computed for $i = 1, 2, ..., l$, and r_l is the hash result for message m [Van Oorschot and Wiener 1999].

2.3. Birthday Paradox

The birthday paradox is the counter-intuitive principle that for groups of as few as 23 persons there is already a chance of about one half of finding two persons with the same birthday (assuming all birthdays are equally likely and disregarding leap years). Compared to finding someone in this group with your birthday where you have 23 independent chances and thus a success probability of $\frac{23}{365} \approx 0.06$, this principle is based on the fact that there are $\frac{23*22}{2} = 253$ distinct pairs of persons. This leads to a success probability of about 0.5 (note that this does not equal $\frac{253}{365} \approx 0.7$ since these pairs are not independently distributed) [Stevens et al. 2012].

2.4. Birthday Attack

The following is the general algorithm for the Birthday Attack and in the next section I will discuss the Birthday Paradox, which is a problem that gave birth to the Birthday Attack algorithm.

1. Let $H : M \rightarrow \{0, 1\}^n$ be a hash function. From this we know that the size of the tag space is $\approx 2^n$ bits and that $|M| \gg 2^n$
2. We choose $2^{\frac{n}{2}}$ random messages in $\mathbb{M}$, i.e. $m_1, m_2, ..., m_{2^{\frac{n}{2}}} \in \mathbb{M}$.
3. For $i = 1, 2, ..., 2^{\frac{n}{2}}$ compute $t_i = H(m_i)$, where t_i is the hash value in the tag space.
4. We then search for any collisions, i.e. $t_i = t_j$ for $i, j \in 1, 2, ..., 2^{\frac{n}{2}}$. If this is not found we go back to step 1 and repeat with different message samples.

2.5. Se pa - Distributed System

A distributed system is a collection of independent computers that appears to its users as a single coherent system [Tanenbaum and Van Steen 2002].

3. Experiments and Evaluation

Foram aplicados tecnicas de paralelismo (openmp) e distribuicao (mpi) visando uma mlehora na performance da busca por colisao. A funcao hash usada nos testes foi a MD5.

4. Conclusions and Future work

Referências

Aiello, W. and Venkatesan, R. (1996). Foiling birthday attacks in length-doubling transformations. In *Advances in Cryptology—EUROCRYPT'96*, pages 307–320. Springer.

Damgård, I. B. (1989). A design principle for hash functions. In *Advances in Cryptology—CRYPTO'89 Proceedings*, pages 416–427. Springer.

Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC press.

Stevens, M. M. J. et al. (2012). *Attacks on hash functions and applications*. Mathematical Institute, Faculty of Science, Leiden University.

Tanenbaum, A. S. and Van Steen, M. (2002). *Distributed systems: principles and paradigms*, volume 2. Prentice hall Englewood Cliffs.

Van Oorschot, P. C. and Wiener, M. J. (1999). Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12(1):1–28.