

# Colliding RFC 3161 Time-Stamp Tokens

– Version 1.0, 4th March 2005<sup>†</sup> –

Alfonso De Gregorio

C&A S.r.l.

V.le Fulvio Testi, 126 – Cinisello Balsamo (MI) 20092, Italy  
alfonso (dot) degregorio (at) acm (dot) org

## Announcement

We extend the method described by Lenstra et al. in [1] to construct pairs of valid, but different, RFC 3161 time-stamp tokens in which the signed data form a collision for the MD5 hash function. This yields to the generation of the same signature by the Time Stamping Authority when it uses MD5 as its hash function.

With this construction the attacker can easily craft MD5 collisions in such a way that he or she constructs RFC 3161 time-stamp tokens in which all fields, except the nonces, are equal (i.e., including the TSA name and serial number). Hence, producing an evidence of the apparent violation, by a given TSA, of an absolute requirement - according to RFC 2119 - of the specifications. In fact, the RFC 3161 requires (see section 2.4.2) the uniqueness of the serial number that, with the TSA name, must identify a unique time-stamp token.

## Construction

The construction is straightforward.

1. First, construct a template for the time-stamp token. All the fields must be completely filled in, with the exception of the nonce and the signature. The following requirements need to be met:
  - (a) the data structure should be compliant to the RFC 3161 standard and the ASN.1 DER encoding rules;
  - (b) the genTime (the time at which the time-stamp token has been created) and the serial-number need to be guessed;
  - (c) the position where the nonce field ends should be an exact multiple of 64 octets.

---

<sup>†</sup> A preliminary version of this announcement was posted the 4th March 2005 on the CFRG Mailing List: "Colliding RFC 3161 time-stamp tokens based on MD5-collisions", Message-ID: <422846C7.4070503@Com-And.COM>

The value at which the `genTime` field will be set by the TSA can be guessed depending on the accuracy of the clock (i.e., the time deviation around the UTC time contained in `genTime`) and the time required to send and consume the time-stamp request message (e.g., the half round-trip latency plus the processing time). The accuracy is a public information and is typically equal to one second, or lesser values.

In several implementations, the serial number is also guessable, since its a monotonically increasing number or derived from the system time.

The third condition can be dealt with by filling out the nonce field with random data up to an exact multiple of 64 octets. This is a first part of the nonce value. For future reference, the offset where the first part of the nonce ends is labeled `Offset-1`. A second bitstring will be appended to it (see the step number 4).

2. The MD5 algorithm get executed on the first portion of the "to be signed" part, truncated at the position `Offset-1` (i.e., where the first part of the nonce ends). The input to MD5 is an exact multiple of 512 bits. The padding normally used in MD5 needs to be suppressed. The output should to be taken as the IV used as input for the next step.
3. Two different bitstring, `b1` and `b2`, whose length is a multiple of 512 bits, get constructed, using the technique developed by Wang et al.[2], for which the MD5 compression function with the IV from the previous step produces a collision.
4. Append the bitstring `b1` into the time-stamp token template after the `Offset-1`. This bitstring completes the nonce value. Now all the (to be) signed data are complete and a time-stamping transaction can start with the given TSA. In the time-stamp request the attacker will specify the nonce that contains the bitstring `b1`.
5. The TSA replies with the time-stamp response that contains (seemingly) the expected time-stamp token.
6. To obtain the second valid time-stamp token, replace `b1` for `b2` in the nonce field. The signature remains valid.

## Validation

The time-stamp tokens are syntactically well-formed and obviously valid also by a cryptographic standpoint, since their digital signature can be verified by relying parties.

The same method may be applied against other iterative hash functions, if a technique is identified to produce collisions with prescribed IV also with the target function.

## Future Work

It would be interesting if someone would be able to construct pairs of colliding RFC 3161 time-stamp tokens whose genTime field, or serial number field, can be arbitrary chosen. This would allow the pre-dating and post-dating of data and would have a severe negative impact on intellectual property protection applications and notary services based on RFC 3161 digital time-stamping - reaffirming the importance of a strong evidence of temporal ordering of time-stamps.

## References

1. Arjen Lenstra, Xiaoyun Wang, and Benne de Weger. Colliding X.509 Certificates. *Cryptology ePrint Archive, Report 2005/067*, 2005, <http://eprint.iacr.org/2005/067>.
2. Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. *Cryptology ePrint Archive, Report 2004/199*, 2004, <http://eprint.iacr.org/2004/199>.