sequential memory-hard key derivation with better measurable security

caveman

January 1, 2021

Abstract

hi — i propose *ciphart*, a sequential memory-hard key derivation function that has a security gain that's measurable more objectively and more conveniently than anything in class known to date.

to nail this goal, *ciphart*'s security gain is measured in the unit of *relative entropy bits*. relative to what? relative to the encryption algorithm that's used later on. therefore, this *relative entropy bits* measure is guaranteed to be true when the encryption algorithm that's used with *ciphart* is also the same one that's used to encrypt the data afterwards.

my reference implementation is available here: `https://github.com/Al-Caveman/ciphart`

Contents

1 ciphart

input:	b	number of entropy bits to be added.
	m_t	memory pad of t^{th} task, at least 32 bytes. initialised to some constant value. $m_t[0:16]$ means first 16 bytes. $m_t[-16:]$ means last 16 bytes.
	R	number of rounds per task.
	e	encryption function.
	k	initial key.
output:	$\hat{k}$	better key.

define P and T such that $PTR - 2^b$ is smallest positive number, $P \geq 2$, T is an even number.
$x \leftarrow 0$ is a 16 bytes wide variable.
for $p = 1, 2, \ldots, P$ **do**
  for $t = 1, 3, \ldots, T-1$, in steps of 2 **do**
    $a \leftarrow t$
    $b \leftarrow t + 1$
    for $r = 1, 2, \ldots, 2R$ **do**
      $n \leftarrow (p, t, r)$
      $m_a \leftarrow e(m_b, k, n)$
      $\hat{a} \leftarrow a$
      $a \leftarrow b$
      $b \leftarrow \hat{a}$
    end for
    $x \leftarrow x \oplus m_a[-16:]$
    $x \leftarrow x \oplus m_b[-16:]$
  end for
  for $t = 1, 2, \ldots, T$ **do**
    $m_t[0:16] \leftarrow m_t[0:16] \oplus x$
  end for
end for
return $\hat{k} \leftarrow hash(m_1, m_2, \ldots, m_T)$

2 parallelism

3 sequential-memory hardness

4 security interpretation

5 comparison

6 summary