

According to the decompilation of the Ciso Vigenere hash algorithm, when the password length is less than 16 the idea behind Ciso Vigenere hash algorithm is:

Let p be the password that the user types.

Let hp be the hardcoded password in the code of Packet Tracer.

Let lp be the length of the user input password.

Let h be the hash value obtained from the custom algorithm.

So that:

$$\begin{aligned}
& \forall h \forall lp \forall hp [(hp = \\
& (d, s, f, d, ;, k, f, o, A, , , ., i, y, e, w, r, k, l, d, J, K, D, H, S, U, B, s, g, v, c, a, 6, 9, 8, 3, 4, n, c, x, v), \\
& 0 < lp < 16, \\
& h_0 = 0, \\
& h_1 = 8, \\
& h = \\
& \Sigma_{i=2}^{lp} \begin{cases} ((p_i \oplus hp_{8+i}) \ggg 4) + 0x30, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xffffffff0 < 0xa0) \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \ggg 4) + 0x37, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xffffffff0 \geq 0xa0) \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \wedge 0xf) + 0x30, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xf < 0x0a) \text{ and if } i \equiv 1 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \wedge 0xf) + 0x37, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xf \geq 0x0a) \text{ and if } i \equiv 1 \pmod{2} \end{cases} \\
&) \implies \#p[p = \mathbf{rev}(h)](0)
\end{aligned}$$

Let's start by proving

$$\begin{aligned}
& \forall h \forall lp \forall hp [(hp = \\
& (d, s, f, d, ;, k, f, o, A, , , ., i, y, e, w, r, k, l, d, J, K, D, H, S, U, B, s, g, v, c, a, 6, 9, 8, 3, 4, n, c, x, v), \\
& 0 < lp < 16, \\
& h_0 = 0, \\
& h_1 = 8, \\
& h = \\
& \Sigma_{i=2}^{lp} \begin{cases} ((p_i \oplus hp_{8+i}) \ggg 4) + 0x30, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xffffffff0 < 0xa0) \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \ggg 4) + 0x37, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xffffffff0 \geq 0xa0) \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \wedge 0xf) + 0x30, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xf < 0x0a) \text{ and if } i \equiv 1 \pmod{2} \\ ((p_i \oplus hp_{8+i}) \wedge 0xf) + 0x37, & \text{if } (h_i \oplus hp_{i+8} \wedge 0xf \geq 0x0a) \text{ and if } i \equiv 1 \pmod{2} \end{cases}
\end{aligned}$$

$$\begin{aligned} &) \implies \nexists p[p = \mathbf{rev}(h)] \\ (0) \end{aligned}$$

I/ substraction to reverse the addition
 $\forall x[(x = y + z) \implies (y = ez)]$ then it follow that as the previous part of the function contains: $h = x + 0x30$, then $h - 0x30 = x$ so
 $\exists rev(h)[rev(H(p)) = p - 0x30]$

II/ exclusive or
 According to the boolean algebra about the exclusive logical or operation,
 $\forall x[(x \oplus x) \implies (x = 0)]$.

Thenas

$xlat \oplus xlat = 0$, and as $p \oplus 0 = p$, we know that the original password
 $p = xlat \oplus h$.

III/ rotating 4 first to 4 last bits
 $\forall x[(x \ggg y) \implies (x \lll y = x)]$.
 Then as $z = (x \ggg y) = (x \lll y)$, we know that the original password
 $p = H(p) \lll 4$.

IV/ unmasking different signatures (recurrent marks) in the hash
 In the previous chapter one ‘I/ substraction to reverse the addition’, we told we can reverse the previous addition. We still need to guess which addition/substraction has been done previously.
 As both addition values are made depending to:
 if $(p_l \wedge 0xf0 < 0xa0) \implies (p_l \wedge 0xf0 + 0x30)$ or else
 $(p_l \wedge 0xf0 > 0xa0) \implies (p_l \wedge 0xf0 + 0x37)$

if $(p_r \wedge 0x0f < 0xa0) \implies (p_r \wedge 0x0f + 0x30)$ or else
 $(p_r \wedge 0x0f > 0xa0) \implies (p_r \wedge 0x0f + 0x37)$

So if the out has the 4 four bits value so that:
 $x \in x|(0xf0x) \leq 0xa0 \implies y = x + 0x30$

So if the out has the 4 four bits value so that:

$$x \in x|(0xf0x) > 0xa0 \implies y = x + 0x37$$

So if the out has the 4 four first bits value so that:

$$x \in x|(0x0fx) \leq 0x0a \implies y = x + 0x30$$

So if the out has the 4 four first bits value so that:

$$x \in x|(0x0fx) > 0x0a \implies y = x + 0x37$$

first byte:

$$\text{as } 0xa0 < 0xf0 + 0x30 < y$$

$$-1 : \forall y \in H(x), x \in x|0xa0 < x \implies [y \in y|0x00 < y < 0xa7]$$

$$-2 : \forall y \in H(x), x \in x|x < 0xa0 \implies [y \in y|0xc0 < y]$$

second byte: as $0xa0 < 0x0f + 0x30 < y$

$$-1 : \forall y \in H(x), x \in x|x < 0x0a \implies [y \in y|0x3a < y]$$

$$-2 : \forall y \in H(x), x \in x|0x0a < x \implies [y \in y|y < 0x4a]$$

Then for both of any subnumber:

$$\forall y = H(x), x \in x|x \leq 0xa \implies y = x + 0x30$$

$$\forall y = H(x), x \in x|x > 0xa \implies y = x + 0x37$$

It follows:

$$\forall y = H(x), y \in y|0 < y \leq 0x0a + 0x30 \implies x = y - 0x30 \text{ then } 0 < x < 0x0a$$

$$\forall y = H(x), y \in y|0 < y \leq 0x0a + 0x37 \implies x = y - 0x30 \text{ then } 0x0a \leq x$$

V /commutativity:

Addition, subtraction and \oplus are commutative.

VI / proof

Then we have already proven each piece of the theorem so that: $hp =$

$$(d, s, f, d, :, k, f, o, A, :, ., i, y, e, w, r, k, l, d, J, K, D, H, S, U, B, s, g, v, c, a, 6, 9, 8, 3, 4, n, c, x, v) \implies$$

$$(\forall x \in hp[0 \geq x0 \geq 256 \implies x \in hp])$$

then:

Let p be the password that the user types.

Let hp be the hardcoded password in the code of Packet Tracer.

Let lp be the length of the user input password.

Let h be the hash value obtained from the custom algorithm.
 So that:

$$\begin{aligned}
 & \forall h \forall l_p \forall h_p [(h_p \in N \wedge 0 \leq h_p, \\
 & 0 < l_p < 16, \\
 & h_0 = 0, \\
 & h_1 = 8, \\
 & h = \\
 & \Sigma_{i=2}^{l_p} \left\{ \begin{array}{ll} ((p_i \oplus h_{p_{i+8}}) \lll 4) - 0x30), & \text{if } h_i < 0xa0 \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus h_{p_{i+8}}) \lll 4) - 0x37), & \text{if } h_i \geq 0xa0 \text{ and if } i \equiv 0 \pmod{2} \\ ((p_i \oplus h_{p_{i+8}}) \wedge 0xfffffffff0) - 0x30), & \text{if } h_i < 0x0a \text{ and if } i \equiv 1 \pmod{2} \\ ((p_i \oplus h_{p_{i+8}}) \wedge 0xfffffffff0) - 0x37), & \text{if } h_i \geq 0x0a \text{ and if } i \equiv 1 \pmod{2} \end{array} \right. \\
 &) \implies \forall p [p = \mathbf{rev}(h)]
 \end{aligned}$$