TIME_CRYPT, A CRYPTOGRAPHIC METHOD THAT ENABLES DECRYPTION BASED ON TIME COMPARISON.

Hammad A. Usmani
Georgia Institute of Technology
Atlanta, Georgia
husmani@gatech.edu

ABSTRACT

Suppose you want to keep something locked only for a certain time or at an exact date and time. You do not want yourself or anyone else to access it until we have reached this time-based requirement. We create a web-based open-sourced (CC0) software as a service (SaaS) where we create a public and private key. Secrets are encrypted, and the SaaS decrypts them based on the stored private key and the valid timestamp. A unique ability of this method is a design that can be used to lock for longer times, in years. Current methods utilizing padlocks and safe products are not designed to withstand physical brute force. In addition, they do not separate concerns between cryptography and durability. The algorithm utilizes industry-standard cryptography and is compatible with locks or safes that can withstand tampering. TIME_CRYPT can integrate with other applications through a representational state transfer (REST) as an application programming interface (API). Experimenters can manually enter a random combination (meant to be forgotten) from the SaaS to set it as a passcode to their physical safes and locks. The current deployment included technologies such as Linux, Python, and PGPy. There are many applications of this ranging from pharmaceuticals, banking, and mental health, among others.

***K*eywords** time · lock · cryptography · time_crypt

1 Introduction

We can set the time lock with a function, f(k,t), where k is a private key that encrypts a message, t is when the time lock expires, and l is a lock or the output of the private key message.

$$f(k,t) = m, l \tag{1}$$

The message contains a random variable that can be represented as a passcode for a safe. It also contains the expiry time in ISO8601 format. Based on the application, the message can have other things.

$$m = (t, p) \tag{2}$$

We only unlock when the current time, c, exceeds the expiry time expressed as an inequality, i.

$$i = c > t \tag{3}$$

We can define another function, f'(), that unlocks the lock. This function utilizes the time comparison or i and returns the unencrypted message with the passcode only if the inequality is true.

$$f'(i, l, k) = m \tag{4}$$