Videocorso PHP

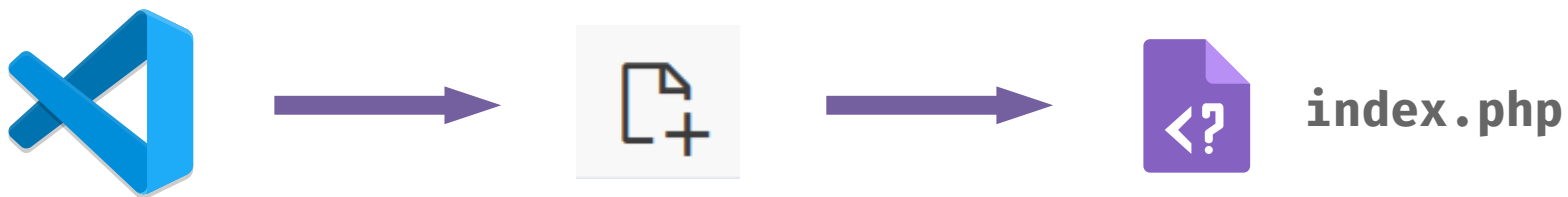
Sicurezza informatica

rendiamo sicure le applicazioni web

Alessandro Flora

Come sarà impostata questa lezione

- All'interno delle slide ci saranno alcuni riferimenti utili per il successivo ripasso
- In particolare nelle slide sono riportati i concetti di base e la sintassi
- La seconda parte sarà un esercizio svolto; è vivamente consigliato provare a ragionare insieme creando uno script per provare in prima persona



Cosa si intende per criptare i dati

- Il verbo *criptare* deriva dal Greco e significa *nascondere*
- Effettivamente andiamo a rendere i dati incomprensibili all'esterno a meno che non si abbiano i dati originali e/o la chiave di decriptazione
- Non è un concetto puramente informatico, la scienza che si occupa di ciò è la matematica ed esiste da ben prima del computer
- Legalmente si è obbligati a farlo per i dati degli utenti e ciò previene fughe di dati legati al furto *fisico* degli stessi o all'accesso alla base di dati

La criptazione dei dati storicamente

- La macchina *Enigma* è un esempio storico di criptazione dei dati
- Venne usata durante la seconda guerra mondiale dalla Germania per comunicare con gli alleati criptando i messaggi
- Serviva un'altra macchina *Enigma* per decriptare i dati ponendo i rotori nelle corrette posizioni
- In totale si arriva a 158 962 555 217 826 350 000 combinazioni possibili dei rotori
- Fu violata da Alan Turing, il padre moderno dell'informatica con la macchina *Bomba*

La crittografia simmetrica

- Quella attuata da *Enigma*
- Consente di criptare e decriptare i dati usando la medesima chiave
- Viene anche chiamata *end-to-end*
- Non consente di verificare se il messaggio è legittimo, basta essere in possesso della chiave unica per codificare e decodificare

La crittografia asimmetrica

- Modello più sicuro e utilizzato nei protocolli di rete
- Sono presenti due chiavi: la privata e la pubblica, con la prima si decripta mentre con la seconda si cripta
- In pratica chiunque può criptare un messaggio per un utente ma solo lui può decriptarlo
- Usando la chiave pubblica del mittente è anche possibile verificarne la provenienza

I protocolli sicuri – HTTPS, FTPS, SMTPS, IMAPS

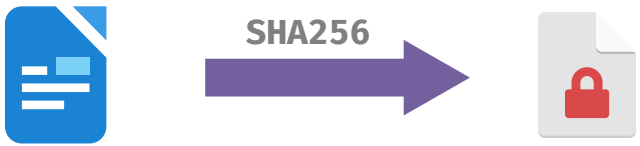
- Sono delle evoluzioni di quelli senza la *S* finale (ovvero HTTP, FTP, SMTP, IMAP)
- Nascono utilizzando il protocollo SSL, attualmente usano il protocollo TLS (v 1.3 – 2018)
- Il protocollo di comunicazione (quelli senza *S*) e quello crittografico (SSL, TLS) sono separati; avviene prima quello crittografico per scambiare le chiavi e stabilire la connessione protetta poi la trasmissione dei dati
- L'evoluzione di TLS chiamata STARTTLS consente di utilizzare anche le porte originarie dei protocolli citati, alternativamente dovevano essere diverse

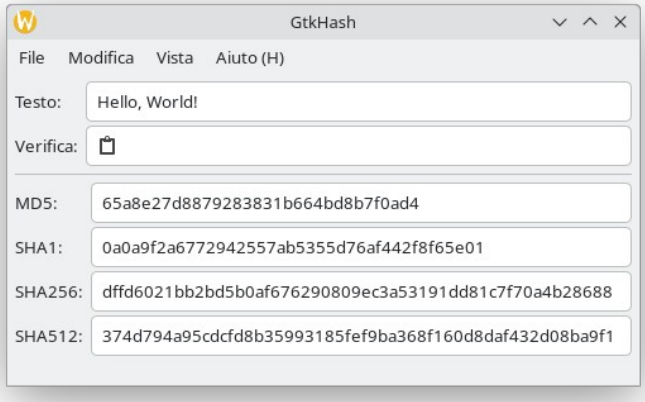
I meccanismi di hash

- Serve avere il dato originario, è una funzione iniettiva ma non invertibile
- Si può facilmente passare dal dato all'hash ma non è possibile il viceversa (a meno di meccanismi di forza bruta ma richiedono troppo tempo)
- Si basano su un meccanismo analogo al resto della divisione che è unico ma non sufficiente per risalire alla divisione originaria
- Sono usati per garantire l'autenticità dei dati scambiati in una comunicazione oppure per salvare informazioni altamente confidenziali che non devono essere decifrabili (per esempio le password)

L'algoritmo SHA256

- Sviluppato dal NSA e brevettato nel 2001, viene però distribuito con licenza libera
- Produce una stringa di 256 bit come output (64 caratteri)
- È possibile verificare l'impronta (*digest*) utilizzando un software quale GTKHash





La crittografia in PHP

- Ci limiteremo alla simmetrica e all'hash
- Useremo funzioni di libreria già integrate
- Servono tendenzialmente per progetti più complessi dove occorre crittografare i dati salvati sulla base di dati
- Sono possibili molteplici approcci, qui vedremo i più semplici

La crittografia simmetrica

- Usiamo le funzioni openssl crypt e decrypt
- Per comodità avete già a disposizione due funzioni già scritte che si occupano di svolgere la criptazione e la decriptazione
- È da usare solamente a scopo didattico, ogni altro uso è **fortemente sconsigliato** e svolto sotto la **propria responsabilità**
- È disponibile a questo indirizzo `https://github.com/AlexF1789/corsoPHP/crypt.php`

Lo script crypt.php

- Dopo averlo richiesto nel proprio script possiamo usare le seguenti funzioni
- Otteniamo rispettivamente una stringa criptata e una decriptata che possiamo salvare in delle apposite variabili

```
cripta_stringa($stringa_originale);

decripta_stringa($stringa_criptata);
```

L’hash tramite SHA256

- Usiamo la seguente funzione passando come secondo argomento il testo da criptare
- Otteniamo una stringa corrispondente all’hash

```
hash('sha256',$testo)
```