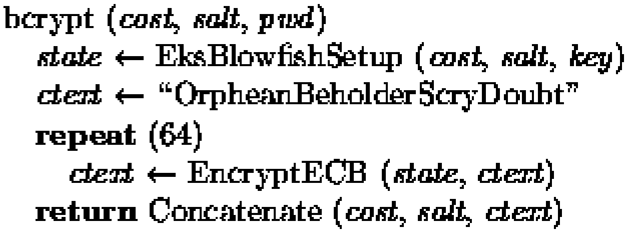
Bcrypt

Bcrypt is a password-hashing function that incorporates a **salt**[1] to protect against rainbow table attacks, which is a precomputed table for caching the output of cryptographic hash functions. Bcrypt is also adaptive because over time the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computational power. Basically, bcrypt being slower makes it better meaning it takes longer for bcrypt to hash the password compared to other hash functions and so cracking the passwords that are stored with bcrypt's hashing function would take a lot **more** time to crack.
Bcrypt storing the passwords as a hash, using a salt to protect against rainbow attacks and being a slow hash function allow for it to be a safe password storage.

We know as technology improves and gets faster, which can also help attackers trying to exploit those systems with brute force attacks and dictionary attacks[2]. Bcrypt also uses "key stretching" which can take a potential bad or short user password and stretch it into a long password/key. 192-bit value OrpheanBeholderScryDoubt. This value is encrypted 64 times using eksblowfish in ECB mode>) with the state from the previous phase. The output of this phase is the cost and the 128-bit salt value concatenated with the result of the encryption loop.



```
bcrypt (cost, salt, pwd)
   state ← EksBlowfishSetup (cost, salt, key)
   ctext ← "OrpheanBeholderScryDoubt"
   repeat (64)
      ctext ← EncryptECB (state, ctext)
   return Concatenate (cost, salt, ctext)
```

Some HIPAA security standards:
Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to authorized persons or software programs.
Encryption and Decryption: Implement a mechanism to encrypt and decrypt electronic protected health information.
Through BCrypt for password storage, we meet these requirements as only authorized users have access to the password hash and BCrypt is one trusted mechanism to encrypt protected health information, in our case user passwords.

"Generated passwords are stored in an encrypted password vault, which can be accessed from multiple devices via a web or mobile app when a master password is entered. Provided a very strong master password is created for the vault... – these solutions are secure and ideal for improving password security in healthcare"(hipaajournal).

Medical places such as OhmniLabs, which does robotic development for healthcare and more, follows HIPAA Compliance for securing patient information and all of their passwords are hashed using Bcrypt
https://ohmnilabs.com/wp-content/uploads/2020/06/HIPAA-Compliance-Guide-20190913.pdf
Doximity is an online networking service for medical professionals that follows Hipaa compliance and all their passwords are salted and hashed using Bcrypt
https://c8y.doxcdn.com/image/upload/v1586884810/Our_Commitment_to_Security_sru46n.pdf
DigitalGator also follows HIPAA compliance and passwords are hashed using Bcrypt
https://www.digitalgator.com/security
Speakap is HIPAA compliant as well using Bcrypt to store passwords as hashes
https://www.speakap.com/en/security-and-compliance

1: a salt is random data that is used as an additional input to a one-way function that hashes data, a password or passphrase.

2: a form of brute force attack used for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying thousands or millions of likely possibilities

Sources:
https://www.hipaaguide.net/hipaa-password-requirements/
https://www.hipaajournal.com/hipaa-encryption-requirements/
https://en.wikipedia.org/wiki/Bcrypt
https://www.npmjs.com/package/bcrypt
https://codahale.com/how-to-safely-store-a-password/
https://www.hipaajournal.com/hipaa-password-requirements/