* **Preventing Adversary from Learning Password Lengths:**
  + To prevent an adversary from learning information about password lengths, we employ padding techniques. Specifically, all passwords are padded to a fixed length before encryption. This ensures that all ciphertexts have uniform lengths, irrespective of the original password lengths. Additionally, encryption algorithms operating in block mode, such as AES-GCM, obscure the exact length of plaintext by padding it to the nearest block size.
* **Preventing Swap Attacks:**
  + Swap attacks are thwarted in our scheme by encrypting each password with a unique key derived from the master password and a site-specific salt. Even if an adversary manages to swap ciphertexts of two passwords, decrypting them without the corresponding master password remains infeasible. Since decryption requires knowledge of the master password, swap attacks are effectively nullified.
* **Necessity of Trusted Location for Rollback Attack Defense:**
  + It is not necessary to assume the existence of a trusted location to defend against rollback attacks. Instead, we can timestamp each update operation and store the timestamp alongside the password record. During password retrieval, the timestamp is checked to ensure retrieval of the most recent version. This approach mitigates rollback attacks without reliance on a trusted location for storing hashes.
* **Handling Lookups with Randomized MACs:**
  + Utilizing randomized MACs instead of HMAC for domain name lookup entails storing the MAC key alongside the domain name. Each lookup operation involves generating a new random key, computing the MAC, and comparing it with the stored MAC. This process incurs a performance penalty due to the need for key generation and MAC computation for each lookup, unlike HMAC where the key remains constant.
* **Reducing Information Leakage about Number of Records:**
  + One approach to reduce information leakage about the number of records involves padding the number of records to a power of 2. For instance, if the actual number of records is kk, padding it to the next highest power of 2 (2⌈log⁡2(k)⌉2⌈log2​(k)⌉) ensures that the number of bits required to represent the padded number of records is always equal to or greater than the number of bits required to represent the actual number of records, thereby concealing the exact number of records from adversaries.
* **Adding Multi-User Support while Maintaining Security:**
  + Multi-user support can be incorporated by encrypting each password with a key derived from both the master password and the user's unique identifier. This ensures that each user can only decrypt passwords encrypted with their specific key. Additionally, access control mechanisms can be implemented to enable users to share access to specific passwords while upholding the security of other passwords. This is achieved by encrypting shared passwords with a separate key shared between authorized users, thus allowing for secure multi-user access.