Google Authenticator

Q.1 Google Authenticator provides strong authentication. What does strong refers to in this case?
Q.2 Assume a browser-based application using Google Authenticator. When prompted, how would you get the verification code?
Q.3 For browser-based applications using Google Authenticator, what does “remember verification for this computer” mean?
Q.4 In Google Authenticator, how long (how many bits) is the shared secret which generates the verification codes, and how is it set up?
Q.5 What is the impact of finding collisions on SHA-1 on the security of Google Authenticator?
Q.6 If an adversary tries verification codes at random, how many attempts does he need before succeeding?
Q.7 What is the protection against automated verification code guessing attacks?
Q.8 What is the difference between HOTP and TOTP? Why do we prefer one to the other?
Q.9 In Google Authenticator, an algorithm uses HMAC on a clock-based value to compute the verification code. How is this clock-based value calculated?
Q.10 How much time is a TOTP verification code valid in Google Authenticator?
Q.11 How can we continue to use Google Authenticator if the smart phone computing the verification code is lost or broken?
Q.12 How to use Google Authenticator for non browser-based applications? What is the advantage compared to authentication without Google Authenticator?
Q.13 Describe a man-in-the-middle attack for a browser-based application using Google Authenticator. How to defeat it?