1.

a. [4 points] Alice wants to send Bob an encrypted e-mail. She generates two ECDH key pairs: one for encryption and one for signing. She uses the **public encryption key** to **encrypt** the e-mail, and attaches a **signature** created by using the **private signing key** to sign a **SHA-256 hash** of the e-mail. The e-mails are e-mail is sent through SMTP. She publicizes both public keys.

   ANS: Bob wouldn't be able to know if the signature is truly from Alice because he is using the public verification key that Alice provided to decrypt the signature. The problem can be solved if Alice and Bob can both be sure that the public verification key is delivered correctly. Pretty Good Privacy (PGP) is a common security program that can be used to decrypt, encrypt, and authenticate emails through digital signatures and file encryption.

b. [4 points] Bob manages a website. To store Alice's password securely, Bob first asks Alice to encrypt it using 128-bit AES with a secret key (the key is shared with Bob). Then, Bob stores a hash of the encrypted version of the password using SHA-512 and also stores the secret key.

   ANS: Passwords should not be encrypted for storage, and we also cannot encrypt and hash passwords at the same time. That is because we would have to check key for all user-logins if user passwords are encrypted, and the action causes the key to be exposed to the login interface as the server needs to key to authenticate. This is dangerous because the key and the password will be stored at the same place, and attackers can steal the key when they can steal the password, which loses the encryption purpose. The correct way to is to just store the hash password into the database, so the login server only checks the hash of the password against the database. The actual password will still need to be sent by the user through an encrypted channel.

c. [4 points] Alice and Bob use the Diffie-Hellman protocol to establish a shared 256-bit secret key. After doing so, Alice wants to send Bob her bank account number so Bob can transfer money to her. Alice encrypts her bank number using the shared 256-bit secret key under AES in counter mode.

   ANS: AES-256 CTR would be a nice block cipher to encrypt Alice's bank number. However, AES in counter mode does not protect message integrity, as attackers can flip arbitrary bits in the encrypted message without decrypting it. This faces insecure message transfer from Alice to Bob, as the message that Bob receives at the end could have been altered, and Bob can never send money to Alice. The integrity issue can be fixed if we combine AES-256 CTR with Message Authentication Code (MAC). By doing so it adds another layer with authenticated encryption against malicious attacker. We would then need both Alice and Bob to share a secret key in order to authenticate identities, so that attackers cannot alter the message without being authenticate.

d. [4 points] In a private end-to-end encrypted messaging app, Alice adds Bob as a friend, which involves downloading his public 512-bit RSA key from a trusted server managing the app. To ensure that the public key is correct, the server uses its private 256-bit ECC key to sign it, which is verified by the corresponding public key that comes with the app. Having Bob's public key allows Alice to start creating secure connections with him.

ANS: 512-bit is too short for RSA keys, which is easy for attackers to brute force break the system. RSA keys need much longer keys, usually 2048/4096 bits. Either Bob has to create a much longer RSA key, or Bob can alter his key establishment method to SKE or AES.

2. [14 points] Cryptography relies on a series of assumptions regarding computational difficulty and key use. In each of the following cases, a commonly held assumption is broken. Discuss the impact of breaking that assumption, focusing on current cryptographic tools, how people interact with them, and how we (including all relevant parties such as cryptosuite developers) should respond to these discoveries.

a. [3 points] A quick polynomial algorithm for integer factorization has been found.

ANS: Then integer factorization is no longer difficult to solve, and the cryptosystems that are based on the difficulty of integer factorization could be easily decrypted by the polynomial algorithm. RSA is an example of public-key cryptosystem that relies on the difficulty of integer factorization. All digital certificates, TLS, web browsers, VPNs, and other communicational channels that uses RSA cryptography would be affected as their encryption is no longer secure, as attackers can decrypt RSA keys with the polynomial algorithm. The easiness to decrypt RSA would affect the confidentiality of the cryptography, as other people's encrypted message could be compromised easier. If such an algorithm is discovered, the products that uses RSA as their public key encryption can consider stop using the algorithm, and instead use Diffie-Hellman for key establishment.

b. [4 points] The private signing key used by Facebook to sign HTTPS certificates was stolen by an unknown group two months ago.

ANS: Facebook's private signing key being compromised could result in attackers impersonating random websites as Facebook certified websites. If the private signing key has already been stolen for two months, it also means that the attackers could have generated multiple valid certificates for different malicious websites signed by Facebook's signing key. People visiting these websites could potentially be attacked or be distributed with malwares. Facebook crypto suite developers finding out this situation should immediately revoke the signed certificates by requesting from the CA. The compromised signing keys should be removed from browsers, and Facebook can generate a new signing key. All of the certificates (that are not actually verified by Facebook) should replace the old certificate with the new certificate signed with the new key.

c. [3 points] An easy way to determine the input corresponding to a given SHA-2 hash has been found.

ANS: SHA-2 refers to the set of cryptographic hash functions with hash values that are 224, 256, 384, or 512 bits. If there exists an easy way to determine the input corresponding to the given SHA-2 hash, it means that attackers can easily reverse the hash to obtain hashed message and passwords. In other words, any system or server that stores hashed password to the database become vulnerable as the data might get leaked. The password database exposed to the login interface is easy to be compromised as it is also exposed to the web. Attackers could steal passwords and decrypt effortlessly. Crypto-suite developers could consider changing the hash functions for hash password storage from SHA-2 to SHA-3, which is a more secure hash function. Developers should also encourage users to update their password in case passwords has already been compromised.

d. [4 points] Large, practical quantum computers have been constructed. (Hint: Start with Shor's algorithm.)

ANS: Construction of large, practical quantum computers means that the Shor's algorithm can be carried out, so that the public-key cryptography schemes can be broken by the algorithm. In other words, any cryptosystem that uses PKE, or involves with prime factorization and discrete logarithmic difficulty is not longer safe to use (i.e. RSA and Diffie-Hellman key exchange). Depending on if the quantum computers can be massively constructed (easiness of construction), the breaking of PKE would have impacted many systems and servers that uses PKE to encrypt and authenticate. Crypto-suite developers should consider transition to SKE cryptography (or block cipher for encryption). However, because PKE is already widely used in many cryptosystems, it would presumably take a long period of time to fully transition away from using PKE.