Contrairement à l'usage plusieurs questions de cet examen sont des questions ouvertes qui admettent plusieurs réponses. Vous devez justifier le choix de vos réponses et détailler les hypothèses que vous adoptez. Vous devez aussi choisir la précision et la longueur de vos réponses pour respecter le temps limité (deux heures) et pour traiter toutes les questions de manière équilibrée.

Exercice 1. GnuPG est un logiciel libre, assez similaire au célèbre PGP, qui permet d'échanger des messages chiffrés.

Extrait de *The GNU Privacy Handbook*, The Free Software Foundation

> Since GnuPG is a hybrid public-key system, the public key is used to encrypt a 128-bit session key, and the private key is used to decrypt it. Key size nevertheless affects encryption and decryption speed since the cost of these algorithms is exponential in the size of the key. Larger keys also take more time to generate and take more space to store. Ultimately, there are diminishing returns on the extra security a large key provides you. After all, if the key is large enough to resist a brute-force attack, an eavesdropper will merely switch to some other method for obtaining your plaintext data. Examples of other methods include robbing your home or office and mugging you. 1024 bits is thus the recommended key size. If you genuinely need a larger key size then you probably already know this and should be consulting an expert in data security.

1. Quels sont les deux types d'algorithmes de chiffrage mentionnés dans ce texte ? Donner des exemples.

2. Associer à chacun des deux types d'algorithmes de chiffrage, chacune des trois clés citées dans le texte.

3. Que représente (mathématiquement) la clé de 1024 bits ? Quelle est la clé privée associée ?

4. Expliquer pourquoi on utilise deux types d'algorithmes de chiffrage différents et donner une esquisse de description de leur utilisation combinée.

5. Ce texte évoque la compléxité du chiffrage avec un algorithme à clé public. Donner un ordre de grandeur de la compléxité du chiffrage avec l'algorithme RSA et une clé de 1024 bits. La deuxième phrase du texte est-elle correcte ?

6. Décrire un algorithme de création d'une clé RSA et analyser sa complexité.

7. Qu'est-ce que la « *brute-force attack* » citée dans le texte ?

8. En vous appuyant sur le texte discuter les relations entre confidentialité et cryptographie.

Exercice 2. On fait référence dans cet exercice aux extraits de la RFC 2409 « The Internet Key Exchange (IKE) » reproduits en annexe. IKE est le protocole d'échange de clé utilisé par IPsec.

1. Rappeler ce qu'est l'algorithme Diffie-Hellman.

2. Pourquoi l'utilisation de `Base Quick Mode` (sans Diffie-Hellman) ne garantit pas le PFS ?

3. Qu'est-ce que la protection contre le rejeu ? Comment est-elle réalisé lors de la phase 2 d'IKE ? Si cette protection n'était pas utilisée que pourrait faire un attaquant ?

4. Qu'est-ce que l'attaque de l'homme-au-milieu ? Est-elle possible dans la phase 2 d'IKE ?

5. La phase 2 d'IKE repose sur la phase 1. Citer au moins deux méthodes permettant

l'identification des protagonistes dans cette phase 1.

6. Combien de SA faut-il pour que deux passerelles fassent communiquer deux sous-réseaux à travers un tunnel IPsec ?

Extraits de la RFC 2409, D. Harkins et D. Carrel, novembre 1998

The Internet Key Exchange (IKE)

3.2 Notation

Nx is the nonce payload; x can be: i or r for the ISAKMP initiator and responder respectively.

prf(key, msg) is the keyed pseudo-random function-- often a keyed hash function-- used to generate a deterministic output that appears pseudo-random. prf's are used both for key derivations and for authentication (i.e. as a keyed MAC). (See [KBC96]).

SKEYID is a string derived from secret material known only to the active players in the exchange.

SKEYID_d is the keying material used to derive keys for non-ISAKMP security associations.

3.3 Perfect Forward Secrecy

When used in the memo Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key. For PFS to exist the key used to protect transmission of data MUST NOT be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys.

5.5 Phase 2 - Quick Mode

Quick Mode is not a complete exchange itself (in that it is bound to a phase 1 exchange), but is used as part of the SA negotiation process (phase 2) to derive keying material and negotiate shared policy for non-ISAKMP SAs. The information exchanged along with Quick Mode MUST be protected by the ISAKMP SA-- i.e. all payloads except the ISAKMP header are encrypted.

Quick Mode is essentially a SA negotiation and an exchange of nonces that provides replay protection. The nonces are used to generate fresh key material and prevent replay attacks from generating bogus security associations. An optional Key Exchange payload can be exchanged to allow for an additional Diffie-Hellman exchange and exponentiation per Quick Mode. While use of the key exchange payload with Quick Mode is optional it MUST be supported.

Base Quick Mode (without the KE payload) refreshes the keying

material derived from the exponentiation in phase 1. This does not provide PFS. Using the optional KE payload, an additional exponentiation is performed and PFS is provided for the keying material.

Quick Mode is defined as follows:

```
     Initiator                              Responder
    -----------                            -----------
     HDR*, HASH(1), SA, Ni [, KE ]  -->
                                    <--   HDR*, HASH(2), SA, Nr [, KE ]
     HDR*, HASH(3)                  -->
```

Where:
HASH(1) is the prf over the message id (M-ID) from the ISAKMP header concatenated with the entire message that follows the hash including all payload headers, but excluding any padding added for encryption. HASH(2) is identical to HASH(1) except the initiator's nonce-- Ni, minus the payload header-- is added after M-ID but before the complete message. The addition of the nonce to HASH(2) is for a liveliness proof. HASH(3)-- for liveliness-- is the prf over the value zero represented as a single octet, followed by a concatenation of the message id and the two nonces-- the initiator's followed by the responder's-- minus the payload header. In other words, the hashes for the above exchange are:

```
HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [ | KE ] )
HASH(2) = prf(SKEYID_a, M-ID | Ni | SA | Nr [ | KE ] )
HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni | Nr)
```

If PFS is not needed, and KE payloads are not exchanged, the new keying material is defined as

```
    KEYMAT = prf(SKEYID_d, protocol | SPI | Ni | Nr).
```

If PFS is desired and KE payloads were exchanged, the new keying material is defined as

```
    KEYMAT = prf(SKEYID_d, g(qm)^xy | protocol | SPI | Ni | Nr)
```

where g(qm)^xy is the shared secret from the ephemeral Diffie-Hellman exchange of this Quick Mode.