14. Donner et expliquer une synthèse sous forme de tableau les différentes possibilités offertes par la cryptologie moderne en termes de confidentialité, intégrité, authentification, non répudiation, échange de clés (utiliser les notations K, PrK, PK, H, S, ...). Pourquoi peut-on souvent distinguer une phase "handshake" et une phase "communication" ? Quel est habituellement le rôle de la cryptographie symétrique et celui de la cryptographie asymétrique ?

Handshake :

Handshake	ALICE	BOB
Login authentifié a)	Envoi du login ———————————————————>	Recupération du login et recherche du pwd dans BD
	Recuperation du nonce <———————————————	Envoit d'un nonce généré
	Génération d'un cNonce	
	Envoi de h =H(nonce + pwd + cNonce) ————->	Recuperation du h
	Envoi du cNonce ———————————————->	Recuperation du cNonce
		Calcule du h sur base du mot de passe de la BD
		h' = H(nonce + pwd + cNonce), si h = h' -> OK
b)	Envoi du login ———————————————————>	Recupération du login, Recherche du pwd dans BD
	Envoi du « salt » ———————————————————>	Recupération du « salt »,
	Calcule du h = H(login + pwd + salt)	Calcule du h' = H(login + pwd + salt)
	Envoir du h ———————————————————->	Récuperation du h, si h = h' -> Ok
Génération des clés	Mise en accord de n et p <———————————	———————————>Mise en accord de n et p
Diffie-Helman : Kab = B^a % p	Envoi de a ———————————————————>	Reception de a
	Reception de b <———————————————	Envoi de b
	Calcule de la clé : Kab = b^a % p	Calcule de Kba = a^b % p
Echange de clés session	Generer Kab -> [Kab]PKb = x	
	Envoi de x ———————————————————>	Reception de x
		Recuperation de PrKb dans un keystore
		[x]PrKb = Kab

Communication :

Communication		ALICE	BOB
Confidentialité :	**a) Symétrique**	{ m } Kab = x ———————————————>	{ x } Kab = m
	b) Asymétrique	Reception de la Pkb (Dans un certificat)<———— { m } Pkb = x ————————————————->	Recuperation de Pkb (dans un certif)et envoit Recuperation de x Recuperation de PrKb dans un keystore { x } PrKb = m
Integrité		Envoi de m ———————————————> Calcule h = H(m) Envoi de h ———————————————>	Reception de m Calcule h' = H(m) Reception de h, si h = h' -> OK
Authentification	**a) Symétrique + Integrité**	Envoi de m ———————————————> H(m + {m}Kab)=HMAC(m) ————————————->	Reception de m Reception de HMAC(m) HMAC'(m) = H(m+{m}Kab) Si HMAC(m) = HMAC'(m) -> OK
	b) Asymétrique Et non répudiation :	Envoi de m ———————————————> {H(m)}Prk = S(m) = s ————————————-> (PrK étant dans un keystore)	Reception de m Reception de s { s }Pka = H(m) = h (Pka étant contenue dans un certificat) h' = H(m) Si h = h' -> OK

Nous pouvons constaté deux type d'échange, tout d'abord une phase de Handshake et puis une de Communication.
La phase de Handshake est indispensable pour authentification du client et puis la génération des clés. En effet, il se peut que n'importe qui ne puisse pas avoir accès au service voulu, nous nous retrouvons donc avec un login avant que le mécanisme de chiffrement se mette en place. Pour ce faire, un simple digest est utilisé. La génération des clés doit être également faite avant la partie communication, si il n'y a pas de clé, pas de chiffrement. Ensuite il ne sert a rien de chiffrer les messages si cette clé passe en claire et puisse être intercepter. Pour ce faire, il y a plusieurs moyen d'échanger les clés. Nous classons c'est échange dans la partie Handshake car ceux-ci sont obligatoire et ne participe pas à une communication active entre les deux cotés.

Ensuite nous constatons la partie communication, qui elle échange les informations, les messages d'un coté a l'autre avec chaque moyen pour vérifiée la l'intégrité / confidentialité / Authentification. Mais à ce stade, toute les informations nécessaire sont connue des deux cotés il n'y a plus de configuration à faire.

La cryptographie symétrique et la cryptographie asymétrique servent tous les deux à échanger des informations par le réseaux sans que ceux-ci puissent être interprété en chemin par une personne tiers. Néanmoins le chiffrement asymétrique est souvent utilisé pour l'échange d'une clé secrète qui permettra un chiffrement symétrique par après. Le chiffrement symétrique étant préfère car celui-ci est plus léger à effectuer (et donc, demande moins de ressource au processeur).

Pour se faire, les deux coté échangeront leurs clé publics (au moyen de certificat) (ou un seul si aucun echange bidirectionnel n'est requit avant le chiffrement symétrique). C'est clé permette de chiffrer un message, mais pas de le décrypter, en effet seule la clé privé associé permet de déchiffrer un message chiffré avec la clé public. Ainsi, un des coté peut générer une clé secrète, la chiffrer avec la clé public de l'autre coté et l'envoyer a ce dernier. L'autre coté utilisera sa clé privé (que seul lui connait), et décryptera la clé secrète. La clé secrète étant connu des deux coté, et de seulement eux, le chiffrement symétrique peut être utilisé des deux cotés.

Chacun des chiffrements à également un avantage. Le chiffrement symétrique permet de faire un hashMac (Combinaison du chiffrement du message et du hashage du message). Ainsi l'autre coté n'aura qu'a faire de même avec le message et sa clé secrète, si il obtient le même résultat, cela veut dire que et le message, et la clé secrète est la même et donc que le message n'a pas été modifié et que c'est bien la bonne personne qui l'a envoyé.
Le chiffrement quand à lui permet de faire des signatures électronique. Une signature est donc crée sur base du message et de la clé privé de la personne émettrice. le coté récepteur vérifiera cette signature avec le message et la clé public du destinataire. Si cela correspond nous avons donc le bon message et cela veut dire que c'est la bonne personne qui l'a envoyé car seul lui connait sa clé privé (qui va avec la clé public que le destinataire a).