TTM4135 Worksheet 7: Key Establishment and Post-Quantum Cryptography

Tjerand Silde and Emil August Hovd Olaisen
{tjerand.silde, emil.august.olaisen}@ntnu.no

Spring 2026

1. Review the definitions of the following concepts:
   (a) Key predistribution;
   (b) Session key distribution;
   (c) Key agreement;
   (d) Kerberos;
   (e) Post-quantum cryptography;
   (f) Learning with errors.
2. Discuss the advantages and disadvantages of using key predistribution, session key distribution and key agreement protocols in the following scenarios:
   - a corporate network such as NTNU's Intranet;
   - a small company or domestic environment;
   - Internet communications (e.g. HTTPS, secure email).

Solution: In the following comparison we assume that key agreement is based on public keys, which requires a supporting public key infrastructure (PKI) to register users, issue certificates, revoke certificates, etc.

- *Corporate network:* Key predistribution is not a good choice for a corporate network because of its dynamic nature. Nodes (e.g. users and hosts) are added regularly. When a new node is added to the network, then every other node in the network needs to be updated with a new key shared with the new node. This process scales poorly.

  Key distribution is generally more adequate here than using PKI + key agreement protocols and is commonly used in practice (e.g. Kerberos in Microsoft networks). The main advantages of key distribution over public-key based key agreement are:

  – key distribution schemes can be based solely on symmetric key algorithms, which are much faster than public key algorithms;
  – no certificate management overhead is incurred.

  The main disadvantage is the need for an online trusted authority (normally referred to as authentication server) which shares a long term secret with each node.

- *Small company or home:* This type of network usually has a small and fairly static membership configuration, i.e. new nodes are added rarely to the network. Hence key predistribution schemes are a good choice. The communication overhead of key distribution is the main disadvantage in this setting.

  This picture may be changing with the Internet of Things making domestic networks more complex and dynamic.

- *Internet:* The most popular choice for the Internet is PKI + key agreement. This is what is used for example in SSL/TLS. Key distribution schemes require all nodes to register with the authentication server in order to establish the long term secret key. This is not feasible in an network as large and diverse as the Internet. Key predistribution is clearly unworkable due to its poor scalability.

3. A potential attack on key establishment protocols is where the attacker makes party A believe that the session key is shared with B but B believes that the same key is shared with C. This is called an *unknown key share attack*.

   - Why might this situation be a serious security problem, even if the attacker does not obtain the key?
   - Why does use of a key derivation function, including the identities of the parties, prevent this attack?

Solution:

- This scenario violates the authentication property we require for key establishment since A and B do not agree who their communication partner is. This may mean, for example, that B would give information, which is intended for C, instead to A. This could be information that B wants to keep confidential from A.
- By including the identities of the intended partner in the KDF, each party can be sure that they will compute the same session key only if they both agree on the intended partner. Note that this assumes that both A and B are honest parties, neither attempting to trick the other. So this just ensures that they will either agree on who they are talking to, or they will compute completely different keys.

4. There are three main approaches to providing *freshness* (protection against replay of messages) in protocols:

   - random challenges,
   - time stamps, and
   - counters.

   (a) Discuss the advantages and disadvantages of each option.

   (b) The fixed Needham-Schroeder protocol in the lecture uses nonces for freshness. Modify it so that freshness is achieved using counters. Specify the checks that each party must perform on receipt of the protocol messages.

Solution:

(a) The goal of all three mechanisms is to assured the receiver of a message that it is *fresh*, i.e. it is not an old message being replayed.

- The first mechanism requires two protocol flows between the sender A and the receiver B. In the first flow, the B sends a newly generated random number, N, to the A. In the second message the sender sends the message M together with the nonce N. On receipt of this, the receiver checks that the nonces are the same.

The main advantage of using random nonces for messages freshness is that the parties do not have to maintain any state across messages. A disadvantage is that it requires interaction between the sender and the receiver. This would exclude applications such as email. Random challenges also require availability of a good random number generator.

- Time stamps only require one flow. A simply sends the message together with current time stamp (a string indicating the time is some agreed format). B checks that the time stamp is recent.

  An advantage of time stamps is that there is no need for interaction between the receiver and sender: it only requires one flow. The main disadvantage is that both A and B must maintained synchronized clocks. Synchronization of clocks is done by interacting with a time server and should be done securely, i.e. in a way that protects against interference form attackers.

- Counters work as follows. The first time A sends a message to B, A sets a counter $C_B = 1$ and sends C_B together with the message. On receipt of the first message from A, B initializes a counter $C_A = 1$. For subsequent messages from A to B, A increases the counter $C_B = C_B + 1$ and attaches it to the message. B then check that $C_B = C_A + 1$, if so it accept the message as fresh and increases $C_A = C_A + 1$.

  As with time stamps, using counters for freshness only requires one flow. A difference with respect to time stamps is that there is no need for a third party (the time server). Disadvantages of counters is that parties must maintain state. Each party must keep a counter for every other party.

(b) In the protocol depicted in Figure 1, $Count_{AS}$ and $Count_{BS}$ are counters shared by the server S with A and B respectively. Both A and B check that received counter is correct, according to their own stored values. Note that there is no need for B to contact A beforehand. A more complex process may be required if synchronization between S and the users may be lost (for example due to interrupted protocol runs).

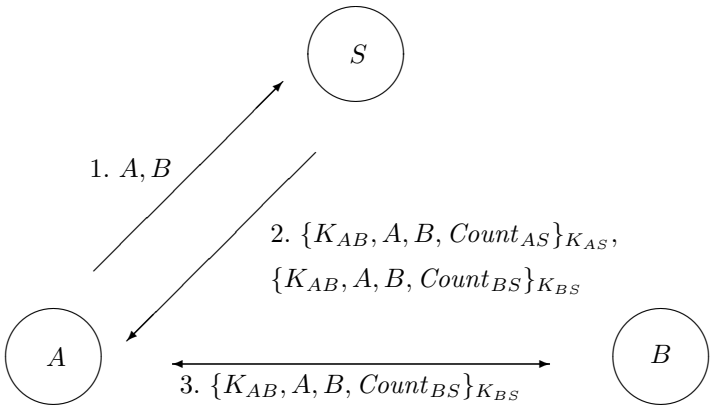

Figure 1: The Needham-Schroeder protocol with a counter instead of nonces.

For the following questions we will use $\mathbf{x} \in [\beta]^n$ to denote a column vector with elements bounded by β.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

with each element $x_i \in \{-\beta, -\beta+1, \ldots, -1, 0, 1, \ldots, \beta-1, \beta\}$. If we have two column vectors $\mathbf{x}, \mathbf{y}$ of length n we can denote their inner-product as:

$$\mathbf{x}^\intercal \mathbf{y} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i.$$

In the following questions, we will consider vectors over the domain $\mathbb{Z}_q$, we will use coefficients from $[-\lfloor q/4 \rfloor, \ldots, \lfloor 3q/4 \rfloor]$. Here $\lfloor \cdot \rfloor$ denotes rounding down. The bound β will be significantly smaller than q, even smaller than $\lfloor q/4 \rfloor$, we can therefore consider $[\beta]$ as a subset of $\mathbb{Z}_q$.

5. Consider the following toy-encryption scheme:

   - KGen, which will output a short secret key $\mathsf{sk} = \mathbf{s} \leftarrow\!\!\$\ [\beta]^n$ and public key $\mathsf{pk} = (\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times n}, \mathbf{t} = \mathbf{A}\mathbf{s})$.
   - $\mathsf{Enc}(\mathsf{pk}, M)$, which on the input of a public key pk and a message $M \in \mathbb{Z}_q$ will sample a random short vector $\mathbf{r} \leftarrow\!\!\$\ [\beta]^n$ and output the ciphertext $c = (\mathbf{u}^\intercal = \mathbf{r}^\intercal \mathbf{A}, v = \mathbf{r}^\intercal \mathbf{t} + M)$.
   - $\mathsf{Dec}(\mathsf{sk}, c)$, which on the input of a secret key sk and ciphertext c will output the message $M' = v - \mathbf{u}^\intercal \mathbf{s}$.

   (a) Show that this scheme is correct, no matter how we choose the parameters n, q, β. That is, show that if $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KGen}$, then, for all n, q, β:

   $$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) = M.$$

   (b) Explain why this scheme is insecure (hint: provide an explicit way to conduct a key recovery attack).

Solution:

(a) The encryption of the message M will be the ciphertext:

$$(\mathbf{r}^\intercal \mathbf{A}, \mathbf{r}^\intercal \mathbf{t} + M).$$

From there we compute the decryption:

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}, (\mathbf{r}^\intercal \mathbf{A}, \mathbf{r}^\intercal \mathbf{t} + M)) &= \mathbf{r}^\intercal \mathbf{t} + M - \mathbf{r}^\intercal \mathbf{A}\mathbf{s} \\ &= \mathbf{r}^\intercal \mathbf{A}\mathbf{s} + M - \mathbf{r}^\intercal \mathbf{A}\mathbf{s} \\ &= M. \end{aligned}$$

(b) Since $\mathbf{A}$ is a square matrix, it is overwhelmingly probable that there exists an inverse matrix $\mathbf{A}^{-1}$. We can therefore compute:

$$\mathsf{sk} = \mathbf{s} = \mathbf{A}^{-1}\mathbf{t}.$$

6. We define $[\beta]$ as the set of integers in the range $[-\beta, \beta]$. Show that if you have two vectors $\mathbf{x}, \mathbf{y} \in [\beta]^n$, then computing the vector multiplication gives an element bounded by $n\beta^2$. Or in other words, show that $\mathbf{x}^\intercal \mathbf{y} \in [n\beta^2]$.

Solution:

We decompose the vectors $\mathbf{x}, \mathbf{y}$ in the normal way: $\mathbf{x}^\intercal = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$. We know that $\mathbf{x}^\intercal \mathbf{y} = \sum_{i=1}^{n} x_i y_i$. Since each $x_i, y_i \in [\beta]$ we know that $x_i y_i \in [\beta^2]$. The sum of n elements from $[\beta^2]$ will give an element in $[n\beta^2]$.

7. Consider the following encryption scheme:

   - KGen, which will output a secret key $\mathsf{sk} = \mathbf{s} \leftarrow\!\!\$\ [\beta]^n$ and public key $\mathsf{pk} = (\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times n}, \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}_1)$, where the noise value $\mathbf{e}_1 \leftarrow\!\!\$\ [\beta]^n$.
   - $\mathsf{Enc}(\mathsf{pk}, M)$, which on the input of a public key pk and a bit-message $M \in \{0, 1\}$ will sample short random values $\mathbf{r}, \mathbf{e}_2 \leftarrow\!\!\$\ [\beta]^n, e_3 \leftarrow\!\!\$\ [\beta]$ and output the ciphertext $c = (\mathbf{u}^\intercal = \mathbf{r}^\intercal \mathbf{A} + \mathbf{e}_2^\intercal, v = \mathbf{r}^\intercal \mathbf{t} + e_3 + m \cdot \lfloor q/2 \rfloor)$.
   - $\mathsf{Dec}(\mathsf{sk}, c)$, which on the input of a secret key sk and ciphertext c will output $M' = v - \mathbf{u}^\intercal \mathbf{s}$.

   Note that this scheme will not output the same plaintext message, but we will show how to extract it below:

   (a) Why does the attack from task 5 not work on this version of the scheme?

   (b)
   - Show that if $M = 0$ then $M' \in [-\beta(2n\beta + 1), \beta(2n\beta + 1)]$.
   - Show that if $M = 1$ then $M' \in [-\beta(2n\beta + 1) + \lfloor q/2 \rfloor, \beta(2n\beta + 1) + \lfloor q/2 \rfloor]$.

   What this means is that if $M = 0$, then M' will be centered around 0, with each coefficient having a maximum distance of $\beta(2n\beta + 1)$ from 0. If $M = 1$ then each coefficient will be centered around $\lfloor q/2 \rfloor$ with each coefficient having a maximum distance of $\beta(2n\beta + 1)$ from $\lfloor q/2 \rfloor$.

   (c) Lastly, show that if $4\beta(2n\beta + 1) < q$, then we can always recover M from M'.

Solution:

(a) If we compute $\mathbf{A}^{-1}\mathbf{t} = \mathbf{s} + \mathbf{A}^{-1}\mathbf{e}_1$ we get a vector that is not equal to $\mathbf{s}$. Since there is no bound placed on $\mathbf{A}^{-1}$, there is no reason that the vector $\mathbf{A}^{-1}\mathbf{e}_1$ should be short, even if $\mathbf{e}_1$ is short.

(b) Given the ciphertext $c = (\mathbf{r}^\intercal \mathbf{A} + \mathbf{e}_2^\intercal, \mathbf{r}^\intercal \mathbf{t} + e_3 + M \cdot \lfloor q/2 \rfloor)$ we compute:

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}, c) &= v - \mathbf{u}^\intercal \mathbf{s} \\ &= \mathbf{r}^\intercal \mathbf{t} + e_3 + M \cdot \lfloor q/2 \rfloor - (\mathbf{r}^\intercal \mathbf{A} + \mathbf{e}_2^\intercal)\mathbf{s} \\ &= \mathbf{r}^\intercal \mathbf{A}\mathbf{s} + \mathbf{r}^\intercal \mathbf{e}_1 + e_3 + M \cdot \lfloor q/2 \rfloor - \mathbf{r}^\intercal \mathbf{A}\mathbf{s} + \mathbf{e}_2^\intercal \mathbf{s} \\ &= \mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 + M \cdot \lfloor q/2 \rfloor. \end{aligned}$$

The output is either $\mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3$ or $\mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 + \lfloor q/2 \rfloor$. Recall that $\mathbf{r}, \mathbf{s}, \mathbf{e}_1, \mathbf{e}_2 \in [\beta]^n$ and $e_3 \in [\beta]$, therefore the term $\mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 \in [\beta(2n\beta + 1)]$. Whether M equals 0 or 1 will determine if the distribution is centered around 0 or $\lfloor q/2 \rfloor$.

(c) Whether M' is in $[-\beta(2n\beta + 1), \beta(2n\beta + 1)]$ or $[-\beta(2n\beta + 1) + \lfloor q/2 \rfloor, \beta(2n\beta + 1) + \lfloor q/2 \rfloor]$ should determine if M is 0 or 1. We would therefore want the two sets to be disjoint, which is equivalent to

$$\begin{aligned} \beta(2n\beta + 1) &< -\beta(2n\beta + 1) + \lfloor q/2 \rfloor \\ 2\beta(2n\beta + 1) &< \lfloor q/2 \rfloor \\ 4\beta(2n\beta + 1) &< 2\lfloor q/2 \rfloor < q. \end{aligned}$$

8. Consider the lattice-based Schnorr-type digital signature scheme in Figure 2. We use r, v, ℓ, n as dimensions and q as a prime. Let β denote a bound, $\bar{\beta}$ denote a slightly larger bound, $\gamma = \ell \cdot \beta$, and let H be a hash function.

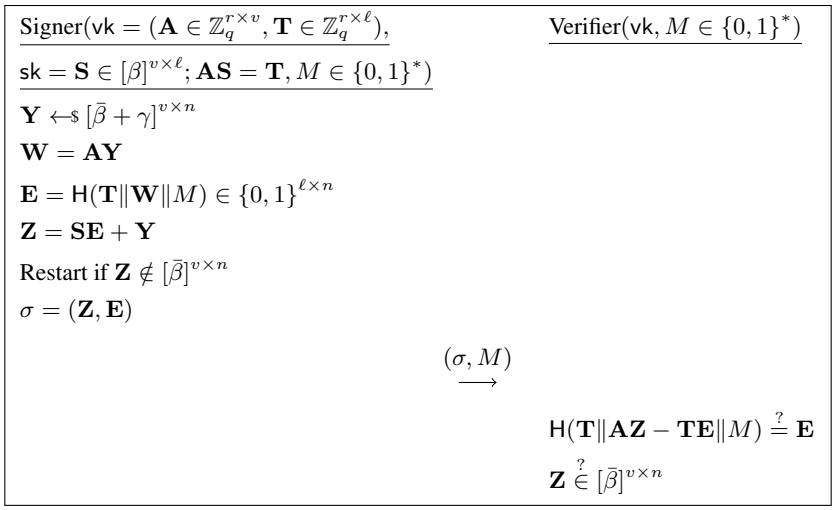

Figure 2: A lattice-based Schnorr-type signature.

- KGen, which will sample random matrices $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{r\times v}, \mathbf{S} \leftarrow\!\$\ [\beta]^{v\times\ell}$ and compute $\mathbf{AS} = \mathbf{T} \in \mathbb{Z}_q^{r\times\ell}$. The algorithm outputs a public verification key $\mathsf{vk} = (\mathbf{A}, \mathbf{T})$ and a secret signing key $\mathsf{sk} = \mathbf{S}$.
- $\mathsf{Sign}(\mathsf{sk}, M)$, which on the input of a signing key sk and an arbitrary message $M \in \{0,1\}^*$ will repeat the following procedure until success (called a rejection sampling procedure):
  1. Sample a random $\mathbf{Y} \leftarrow\!\$\ [\bar{\beta} + \gamma]^{v\times n}$ and compute $\mathbf{W} = \mathbf{AY}$.
  2. Set $\mathbf{E} = \mathsf{H}(\mathbf{T}\|\mathbf{W}\|M) \in \{0,1\}^{\ell\times n}$ and compute $\mathbf{Z} = \mathbf{SE} + \mathbf{Y}$.
  3. Restart if $\mathbf{Z} \notin [\bar{\beta}]^{v\times n}$.

  Finally, output signature $\sigma = (\mathbf{Z}, \mathbf{E})$ together with the message M.
- $\mathsf{Vf}(\mathsf{vk}, \sigma, M)$, which will output 1 if and only if $\mathsf{H}(\mathbf{T}\|\mathbf{AZ} - \mathbf{TE}\|M) = \mathbf{E}$ and $\mathbf{Z} \in [\bar{\beta}]^{v\times n}$.

(a) Show that the signature scheme is correct. That is, show that if $(\mathsf{vk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KGen}$ then:

$$\mathsf{Vf}(\mathsf{vk}, \mathsf{Sign}(\mathsf{sk}, M), M) = 1.$$

(b) Explain why $\mathbf{Z} = \mathbf{SE} + \mathbf{Y}$ is always contained in $[\bar{\beta} + 2\gamma]^{v\times n}$ (before the rejection step). Furthermore, if we assume that $\mathbf{Z}$ is uniformly distributed in $[\bar{\beta} + \gamma]^{v\times n}$, then the probability of success in the rejection sampling step (meaning that $\mathbf{Z} \in [\bar{\beta}]^{v\times n}$) of Sign is:

$$\Big(\frac{2\bar{\beta} + 1}{2(\bar{\beta} + 2\gamma) + 1}\Big)^{v\cdot n}.$$

How is this success probability of the rejection sampling impacted by the size of $\bar{\beta}$? Why is this rejection sampling step important for efficiency and security?

(c) Consider these two potential adversarial powers:

1. Given the public verification key $\mathsf{vk} = (\mathbf{A}, \mathbf{T})$, find a matrix $\mathbf{S}' \in [\beta]^{v\times\ell}$ such that $\mathbf{AS}' = \mathbf{T}$.
2. Given matrices $\mathbf{W}' \in \mathbb{Z}_q^{r\times n}$ and $\mathbf{E}' \in \{0,1\}^{\ell\times n}$, find $\mathbf{Z}' \in [\bar{\beta}]^{v\times n}$ such that $\mathbf{AZ}' = \mathbf{W}' + \mathbf{TE}'$.

What type of attacks could these adversaries do? Is one of these problems harder than the other? Explain why both of these problems need to be hard for the scheme to be secure.

(d) How does the size of the parameters β and $\bar{\beta}$ impact the hardness of the lattice problems (c)?

Solution:

(a) The verifier first checks that $\mathsf{H}(\mathbf{T}\|\mathbf{AZ} - \mathbf{TE}\|M) = \mathbf{E}$, which will happen if $\mathbf{AZ} - \mathbf{TE} = \mathbf{W}$:

$$\mathbf{AZ} - \mathbf{TE} = \mathbf{A}(\mathbf{SE} + \mathbf{Y}) - \mathbf{ASE} = \mathbf{AY} = \mathbf{W}.$$

The second check is that $\mathbf{Z} \in [\bar{\beta}]^{v \times n}$, this is ensured since the signer will sample $\mathbf{Y}$ until $\mathbf{Z} \in [\bar{\beta}]^{v \times n}$.

(b) We need to check where $\mathbf{SE}$ lies. Each term in the matrix $\mathbf{SE}$ is an inner product of binary vector of length ℓ and a vector with coefficients in $[\ell \cdot \beta] = [\gamma]$, thus the matrix $\mathbf{Z}$ is in $[\bar{\beta} + 2\gamma]^{v \times n}$. This set has $(2(\bar{\beta} + 2\gamma) + 1)^{v \cdot n}$ elements and the size of $[\bar{\beta}]^{v \times n}$ is $(2\bar{\beta} + 1)^{v \cdot n}$. The probability of sampling inside of $[\bar{\beta}]^{v \times n}$ is then:

$$\Big(\frac{2\bar{\beta} + 1}{2(\bar{\beta} + 2\gamma) + 1}\Big)^{v \cdot n},$$

if we assume uniform distribution of $\mathbf{Z}$ inside of $[\bar{\beta} + 2\gamma]^{v \times n}$. The larger $\bar{\beta}$, the larger the success probability is, however, this leads to larger signatures. The goal is to find a balance between the running time (how many times one rejects on average) and the size of the signatures (how big the masking value is). The rejection sampling step is important to avoid that the response $\mathbf{Z}$ leaks any information about the secret $\mathbf{S}$, so it is important that it is of a size that it is independent of $\mathbf{S}$.

(c) Solving the first problem will give the adversary the power to obtain a valid signing key sk' for the verification key vk. This may not be the same as the one from the signer, but it still possesses the same properties as that key. It is a key recovery attack.

Solving the second problem gives the adversary a signature $(\mathbf{Z}', \mathbf{E}')$ for the message M when given $\mathbf{W}'$ and $\mathbf{E}' = \mathsf{H}(\mathbf{T}\|\mathbf{W}'\|M)$. It is a signature forgery attack.

We remark that both of these problems are instantiations of SIS where we are supposed to find matrices with short entries. We know from the lectures that the SIS problem gets easier when the allowed coefficients gets larger, so it is harder to find a solution that must be bounded by β vs. a solution bounded by a larger bound $\bar{\beta}$ (if the dimensions are roughly the same).

Further, we note that if we find a solution to the first problem, then we can also solve the second problem by following the signing algorithm. Hence, the first problem is harder than the second.

It is crucial for a signature scheme to be secure that it is impossible to conduct key recovery and signature forgery attacks, so we need both of these problems to be hard.

(d) The smaller β is, the easier the Learning With Errors problem is, but this can be compensated with higher dimension of the matrices. It is important for security that the public key looks like a uniformly random vector. When β is small, then the Short Integer Solutions problem is hard.

The larger $\bar{\beta}$ is, the easier the Short Integer Solutions problem is, but this can also be compensated with higher dimension of the matrices. It is important for security that the signatures are small enough so that it is hard to find forgeries.

While encryption only relies on the Learning With Errors problem, signatures rely on both of them to be hard at the same time, making it a bit more challenging to find secure parameters.