Practical Post-Quantum Hierarchical Identity-Based Encryption

Peter Campbell, NCSC
Michael Groves, NCSC
Identity-based encryption

Identifier $ID_U$

Bob

Alice
Identity-based encryption

\[(A, a) = \text{MasterKey}()\]

KMS

Identifier \(ID_U\)

Bob

Alice
Identity-based encryption

\[(A, a) = \text{MasterKey}( )\]

\[x = \text{Extract}_a(ID_U)\]
Identity-based encryption

\[(A, a) = \text{MasterKey}()\]
\[x = \text{Extract}_a(ID_U)\]

\[Z = \text{Enc}_{A, ID_U}(M)\]

Alice
Identifier \(ID_U\)
\[M = \text{Dec}_x(Z)\]

Bob

\[x = \text{Extract}_a(ID_U)\]

KMS
Hierarchical identity-based encryption

Root KMS

Delegated KMS

Identifier $ID_K$

Identifier $ID_U$

Bob

Alice
Hierarchical identity-based encryption

\[(A, a) = \text{MasterKey}(\ )\]

Root KMS

Delegated KMS

Identifier \(ID_K\)

Identifier \(ID_U\)

Alice

Bob

Root KMS

Delegated KMS

Identifier \(ID_K\)

Identifier \(ID_U\)

Alice

Bob
Hierarchical identity-based encryption

\[(A, a) = \text{MasterKey()}\]
\[s = \text{Delegate}_{a}(ID_K)\]
Hierarchical identity-based encryption

\[ (A, a) = \text{MasterKey}(\; ) \]

\[ s = \text{Delegate}_a(ID_K) \]

Root KMS

\( ID_K \)

Delegated KMS

\[ x = \text{Extract}_s(ID_U) \]

Identifier \( ID_K \)

Identifier \( ID_U \)

Bob

Alice

This information is exempt under the Freedom of Information Act 2000 (FOIA) and may be exempt under other UK information legislation. Refer any FOIA queries to ncscinfoleg@ncsc.gov.uk
Hierarchical identity-based encryption

\[(A, a) = \text{MasterKey}(\ )\]
\[s = \text{Delegate}_a(ID_K)\]

Root KMS

Delegated KMS

\[Z = \text{Enc}_{A, ID_K||ID_U}(M)\]

Bob

Alice

Identifier \(ID_K\)

\[x = \text{Extract}_s(ID_U)\]

\[M = \text{Dec}_x(Z)\]
Motivation

Identity-based encryption can offer a lightweight option for key management in enterprise applications where a full public-key infrastructure is impractical or unnecessary.

Example applications include:

• Encrypted e-mail;
• Secure voice;
• Internet of Things.

The UK Government is already using SAKKE [1], a variant of the pairing-based scheme by Sakai and Kasahara [2], and it is due to be adopted by UK emergency services from 2018.
Hierarchical identity-based encryption over lattices

Ducas-Lyubashevsky-Prest [3]:
• Identity-based encryption scheme using NTRU-style lattices.
• Practical (see Sarah’s talk on DLP tomorrow [4]) but not hierarchical.

Cash-Hoffheinz-Kiltz-Peikert [5]:
• General approach to hierarchical identity-based schemes using Bonsai trees.
• Standard lattices only so ciphertext sizes are substantial.
Hierarchical identity-based encryption over lattices

Ducas-Lyubashevsky-Prest [3]:
• Identity-based encryption scheme using NTRU-style lattices.
• Practical (see Sarah’s talk on DLP tomorrow [4]) but not hierarchical.

Cash-Hoffheinz-Kiltz-Peikert [5]:
• General approach to hierarchical identity-based schemes using Bonsai trees.
• Standard lattices only so ciphertext sizes are substantial.

LATTE:
• Hierarchical scheme obtained by adapting the Bonsai trees to NTRU-style lattices.
• Reuses many of the techniques from DLP (see also Thomas’s talk on Falcon [6]).
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_q^{n \times n}, \quad b \in \mathbb{Z}_q^n \]

Lattice

\[ \mathcal{L}(A) = \{ x \in \mathbb{Z}^{2n} : [I \mid A]x = 0 \in \mathbb{Z}_q^n \} \]
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_q^{n \times n}, \ b \in \mathbb{Z}_q^n \]

Lattice

\[ \mathcal{L}(A) = \{ x \in \mathbb{Z}^{2n} : \begin{bmatrix} I \ | \ A \end{bmatrix}x = 0 \in \mathbb{Z}^n \} \]

\[ \mathcal{L}(A || A_K) = \{ x \in \mathbb{Z}^{3n} : \begin{bmatrix} I \ | \ A \ | \ A_K \end{bmatrix}x = 0 \in \mathbb{Z}_q^n \} \]

\[ A_K = H(ID_K) \]
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_{q}^{n \times n}, \ b \in \mathbb{Z}_{q}^{n} \]

Lattice

\[ \mathcal{L}(A) = \{ x \in \mathbb{Z}^{2n} : [I \parallel A]x = 0 \in \mathbb{Z}^{n} \} \]

\[ A_{K} = H(ID_{K}) \]

\[ \mathcal{L}(A \parallel A_{K}) = \{ x \in \mathbb{Z}^{3n} : [I \parallel A \parallel A_{K}]x = 0 \in \mathbb{Z}^{n}_{q} \} \]

\[ A_{U} = H(ID_{K} \parallel ID_{U}) \]

\[ \mathcal{L}(A \parallel A_{K} \parallel A_{U}) = \{ x \in \mathbb{Z}^{4n} : [I \parallel A \parallel A_{K} \parallel A_{U}]x = 0 \in \mathbb{Z}^{n}_{q} \} \]
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_{q}^{n \times n}, \quad b \in \mathbb{Z}_{q}^{n} \]

Lattice

\[ \mathcal{L}(A) \]

Private key

Short basis \( S \) for \( \mathcal{L}(A) \)

\[ A_{K} = H(ID_{K}) \]

\[ \mathcal{L}(A \parallel A_{K}) \]

\[ A_{U} = H(ID_{K} \parallel ID_{U}) \]

\[ \mathcal{L}(A \parallel A_{K} \parallel A_{U}) \]
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_q^{n \times n}, \quad b \in \mathbb{Z}_q \]

Lattice

\[ \mathcal{L}(A) \]

Private key

Short basis \( S \) for \( \mathcal{L}(A) \)

\[ \text{Delegate}(ID_K) \]

\[ \mathcal{L}(A || A_K) \]

Short basis \( S_K \) for \( \mathcal{L}(A || A_K) \)

\[ A_K = H(ID_K) \]

\[ A_U = H(ID_K || ID_U) \]

\[ \mathcal{L}(A || A_K || A_U) \]
Bonsai trees

Public Key

\[ A \in \mathbb{Z}_q^{n \times n}, \quad b \in \mathbb{Z}_q^n \]

Lattice

\[ \mathcal{L}(A) \]

Private key

\[ A_K = H(ID_K) \]

\[ \mathcal{L}(A || A_K) \]

Short basis \( S \) for \( \mathcal{L}(A) \)

\[ A_U = H(ID_K || ID_U) \]

\[ \mathcal{L}(A || A_K || A_U) \]

Short basis \( S_K \) for \( \mathcal{L}(A || A_K) \)

Delegate\((ID_K)\)

\[ b + \mathcal{L}(A || A_K || A_U) \]

Extract\((ID_U)\)

Short vector \( x \) in the coset
Our hierarchical identity-based encryption scheme uses the same setup as in [3]:

- $n$ is a power of two;
- $q$ is a prime with $q \equiv 1 \pmod{2n}$.

These define the cyclotomic rings

$$R = \mathbb{Z}[x]/(x^n + 1) \quad \text{and} \quad R_q = R/qR.$$ 

We will also need three different discrete Gaussian distributions $\chi_0, \chi_1, \chi_2$ over $R$. Their standard deviations will depend on $n$ and $q$. 
Generation

The root KMS samples $a_1, a_2 \in R$ from $\chi_0$ and then tries to find small $a_1', a_2' \in R$ with

$$a_1a_2' - a_1'a_2 = q.$$ 

If this is not possible, or if $a_2$ is not invertible mod $q$, then it tries again. Otherwise,

$$S = \{(a_1, a_2), (a_1', a_2')\}$$

will be a short private basis for the lattice $\mathcal{L}(A)$ corresponding to the public element

$$A = -a_1/a_2 \in R_q.$$ 

The root KMS samples the additional public element $B \in R_q$ uniformly at random.
Delegation

The root KMS uses the master private basis $S$ in the randomised nearest plane algorithm from [3] to find $s_1, s_2, s_3 \in R$ and $s'_1, s'_2, s'_3 \in R$ such that

\[
\begin{align*}
    s_1 + As_2 + A_K s_3 &= 0 \\
    s'_1 + A's'_2 + A_K s'_3 &= 0
\end{align*}
\]

where the $s_i$ and $s'_i$ appear to have been sampled from $\chi_1$. 
Delegation

The root KMS uses the master private basis $S$ in the randomised nearest plane algorithm from [3] to find $s_1, s_2, s_3 \in R$ and $s'_1, s'_2, s'_3 \in R$ such that

\[
\begin{align*}
    s_1 + As_2 + A_K s_3 &= 0 \\
    s'_1 + As'_2 + A_K s'_3 &= 0
\end{align*}
\]

where the $s_i$ and $s'_i$ appear to have been sampled from $\chi_1$.

It then tries to find small elements $s''_1, s''_2, s''_3 \in R$ such that

\[
    s''_1 (s_2 s'_3 - s'_2 s_3) - s''_2 (s_1 s'_3 - s'_1 s_3) + s''_3 (s_1 s'_2 - s'_1 s_2) = q.
\]

The delegated basis for $L(A || A_K)$ will be $S_K = \{(s_1, s_2, s_3), (s'_1, s'_2, s'_3), (s''_1, s''_2, s''_3)\}$. 
Extraction

The delegated KMS uses the delegated basis $S_K$ in the randomised nearest plane algorithm to find small $x_1, x_2, x_3, x_4 \in R$ such that

$$x_1 + Ax_2 + x_3 A_K + x_4 A_U = B$$

where the $x_i$ appear to have been sampled from $\chi_2$. 
Extraction

The delegated KMS uses the delegated basis $S_K$ in the randomised nearest plane algorithm to find small $x_1, x_2, x_3, x_4 \in R$ such that

$$x_1 + Ax_2 + x_3A_k + x_4A_U = B$$

where the $x_i$ appear to have been sampled from $\chi_2$.

To avoid leaking information, the standard deviations should be chosen so that

$$\sigma(\chi_0) = O\left(\sqrt{q/n}\right), \quad \sigma(\chi_1) = O(\sqrt{q}), \quad \sigma(\chi_2) = O(\sqrt{qn}).$$

(See Thomas’s talk on Thursday for more details, hopefully.)
Encryption

Encryption and decryption are fairly standard, except that they involve more terms, and they follow the one-to-many-bit encoding approach by Pöpplemann and Güneysu [7].

To encrypt, Bob chooses $e_1, e_2, e_3, e_4, e_5 \in R$ with coefficients in $\{-1,0,1\}$ and forms

\[
V_1 = Ae_1 + e_2, \quad V_2 = A Ke_1 + e_3, \quad V_3 = A Ke_1 + e_4, \quad V_4 = Be_1 + e_5 + \text{Encode}(m).
\]

To decrypt, Alice forms

\[
W = V_4 - V_1 x_2 - V_2 x_3 - V_3 x_4 = e_1 x_1 - e_2 x_2 - e_3 x_3 - e_4 x_4 + e_5 + \text{Encode}(m)
\]

and recovers $m = \text{Decode}(W)$. 
Encryption

Encryption and decryption are fairly standard, except that they involve more terms, and they follow the one-to-many-bit encoding approach by Pöpplemann and Güneysu [7].

To encrypt, Bob chooses $e_1, e_2, e_3, e_4, e_5 \in R$ with coefficients in $\{-1,0,1\}$ and forms

$$V_1 = Ae_1 + e_2, \quad V_2 = A_K e_1 + e_3, \quad V_3 = A_K e_1 + e_4, \quad V_4 = B e_1 + e_5 + \text{Encode}(m).$$

To decrypt, Alice forms

$$W = V_4 - V_1 x_2 - V_2 x_3 - V_3 x_4 = e_1 x_1 - e_2 x_2 - e_3 x_3 - e_4 x_4 + e_5 + \text{Encode}(m)$$

and recovers $m = \text{Decode}(W)$.

Note that it is important to use an actively secure transform such as Fujisaki-Okamoto [8].
References


Questions?