Anonymity from Asymmetry

Some new constructions of anonymous HierarchicalIBE systems using asymmetric pairing, without random oracle

Introduction
In an Identity-Based Encryption system (IBE), any string can function as a public key. A master secret is used to generate private keys for any public-key of interest.

Alice only needs to remember PP to Encrypt messages for anyone.

An extension of IBE, called Hierarchical-IBE, allows for a hierarchy of identities where any path from the root to a node can function as a public key.

System Primitives
Setup: given bilinear groups (G, G) with generators (g, g), select a random (\alpha, \beta, \gamma, \delta) \in \mathbb{Z}_p, and set: g_0 = g^\gamma, g_{\alpha} = g^\beta, h = g_\delta, f = g^\alpha, t = g^\gamma, and their analogues: g_{\beta} = g^\alpha, g_{\delta} = g^\beta.
Then:
PP = (\alpha, \beta, \gamma, \delta, g_0, g_\alpha, g_{\beta}, g_{\delta}, g, f, t, h) \in G^2 \times \mathbb{G}^2
mk = (g_\alpha, g_{\beta}, g_\delta, g, h, f, t) \in \mathbb{G}^2

Encrypt(PP, ID, M): To encrypt a message M \in \mathbb{G}_1 under the public key ID = 1 \in \mathbb{Z}_n, pick a random \epsilon \in \mathbb{Z}_n, and output:
C = (M \cdot (g_\epsilon, g_\epsilon), g_\epsilon, (f_\epsilon, t_\epsilon)) \in \mathbb{G}_1 \times \mathbb{G}_2

Decrypt((C, \epsilon)): To decrypt a ciphertext C = (A, B, \epsilon, C) \in \mathbb{G}_1 \times \mathbb{G}_2 using the private key \epsilon = (d_\alpha, d_\beta, d_\delta) \in \mathbb{G}_1,
output:
A \cdot \epsilon(C_1, d_\alpha) \cdot \epsilon(C_2, d_\beta) \cdot \epsilon(B, d_\delta) \in \mathbb{G}_1

Security proof
In BHE, only \epsilon_0 had to be kept secret. It is hard to guess from calls to Extract, thanks to the randomness introduced by \epsilon_1, \epsilon_2, \epsilon_3. Security proof of our new system requires a series of games. Intuitively, we prove that Extract reveals no information about the master key mk.

Extend to Hierarchical IBE
To extend BHE, IBE in an IBE, we use different \epsilon for each level of the hierarchy: \epsilon_1, \epsilon_2, \epsilon_3. Now, we want those values \epsilon_1 to be kept secret to insure anonymity, but are not needed to delegate to children in the hierarchy tree. We solve this dilemma by giving \epsilon_1, but blinded in such a way that we can use it to delegate, by some linear algebra operation.

An Extension: delegatable HIBE—
One of our construction, can be extended to a delegatable HIBE system, a particularly useful system for search in encrypted data.

Informally, in an HIBE, message are encrypted depending on a vector of properties, vector that must also be hidden. Keys correspond to similar vectors, with an additional wildcard + that can replace some properties. The description of messages must be possible with the key If it and only if their properties match everywhere but the wildcard +. Delegation means that one can build a key \epsilon from a key \epsilon and if, \epsilon is not \epsilon and only if \epsilon \epsilon. The HIBE may be seen as a restriction of a Delegatable HIBE system, where we only ask for keys having a vector of properties the form (\epsilon_1, \epsilon_2, \epsilon_3, ...).