Inner product enctyption

by K.Krutoy

Feb 2020

Content

- For what?
- Modules
- Hamming Distance
- What is going on?
- Modification

For what?

Why do we need to care about biometrics and homomorphic encryption?

- Once your biomeric data is stolen - never you'll be able to change it. Thus we need to invent Registration-Authentication systems that will prevent it. Homomorphic encryption gives us an opportunity to know how close our authentication data to registration data without decryption.

Modules

- An important topic is modules over commutative rings, in our case we'll use finite fields ($\mathbb{F}_{2^{256}}$). One can read what is it on Wikipedia. In particular we are interested in cyclic modules over $\mathbb{F}_{2^{256}}$

Hamming Distance

- HD(x,y) $= \frac{dim-(x,y)}{2dim}$

What is going on? (1)

Abit about IPE algorithm

Suppose we have three cyclic $\mathbb{F}_{2^{256}}$-modules: $\hat{G_1}, \hat{G_2}, \hat{G_T}$, where first two of them are using additive notation and the last uses multiplicative one. Also we have a map (so called pairing) $e : \hat{G_2} \times \hat{G_1} \to \hat{G_T}$ with such properties:

- $\forall a, b \in \mathbb{F}_{2^{256}} : e(aG_2, bG_1) = e(G_2, G_1)^{ab}$
- $e(G_2, G_1) \neq 1$
- computability

We will take vectors of even dimension (we'll call it dim) which elements are -1,1. We can compute their Hamming Distance, but we want to compute it while they are encrypted.

What is going on? (2)

The algorithm

Here we have four steps: key generation, registration, authentication, hamming distance

- KeyGen returns a specific list of random parameters (Master key)
- Registration takes msk and the registration vector and returns a registration template, that is a vector of length dim+4 with elements in $\hat{G}_2$ that are specifically constructed.
- Authentication takes msk and the authentication vector and returns an authentication template, that is a vector of length dim+4 with elements in $\hat{G}_1$ that are specifically constructed.
- Hamming distance will take reg template and auth template to do the following: $\chi = \prod_{i=1}^{dim+4} e(regt_i, autht_i) = e(G2, G1)^{\sum_i regt_i autht_i}$ After what it will return the discrete logarithm of χ. Due to the specific calculations in Registration and Authentication the thing will be equals to Hamming Distance times dim

The problem my modification fixes is the predictable behavior of coefficients $regt_i, autht_i$. We can calculate $e(G2, G1)^{regt_i autht_i}$, take the discrete logarithm and understand if i's bits of our vectors are the same or not.

Modification

The algorithm

To prevent this we'll generate a random non-singular matrix over $\mathbb{F}^{256}$ and act on $regt_i$ by it's inverse and on $autht_i$ by it's transpose, the thing won't change the value of the $\sum_i regt_i auth_i$

We'll call $autht_i, regt_i$ by a_i, r_i respectively. Suppose we have the following:

$$a_i \to \sum_j \alpha_{j,i} a_j, r_i \to \sum_j \beta_{i,j} r_j$$

$$\sum_k r_k a_k = \sum_k (\sum_j \alpha_{j,k} a_j \times \sum_i \beta_{k,i} r_i) =$$

$$\sum_k \sum_i \sum_j \alpha_{j,k} \beta_{k,i} a_j r_i = \sum_i r_i \sum_k \sum_j \alpha_{j,k} \beta_{k,i} a_j = \sum_i r_i a_i$$

Then it follows that: $\sum_j a_j \sum_k \alpha_{j,k} \beta_{k,i} = a_i$, then $\sum_k \alpha_{j,k} \beta_{k,i} = \delta_{i,j}$ where $\delta_{i,j}$ is the Kronecker's delta. So

$$[\alpha_{j,i}]_{i,j=1}^{dim+4} = inv([\beta_{i,j}]_{i,j=1}^{dim+4})$$

must hold to preserve saclar product.

The End