Notes on threshold Diffie-Hellman cryptosystem

Ganyu Xu[1]

University of Waterloo, Ontario, Canada
g66xu@uwaterloo.ca

1 TDH0a: naive threshold Diffie-Hellman cryptosystem

Let (G, g) be a cyclic group of prime order q. Let (E, D) be a symmetric cipher with key space $\mathcal{K}$. Let $H : G \rightarrow \mathcal{K}$ be a hash function. Consider the following ElGamal-like cryptosystem. We will call this `ElGamal`:

Algorithm 1 $\texttt{ElGamal.KeyGen}()$

1: $\alpha \leftarrow \mathbb{Z}_q$
2: $\texttt{sk} \leftarrow \alpha$
3: $\texttt{pk} \leftarrow g^\alpha$
4: **return** $(\texttt{pk}, \texttt{sk})$

Algorithm 2 $\texttt{ElGamal.Enc}(\texttt{pk} = g^\alpha, m)$

1: $\beta \leftarrow \mathbb{Z}_q$
2: $v \leftarrow g^\beta$
3: $w \leftarrow (g^\alpha)^\beta$
4: $k \leftarrow H(w)$
5: $s \leftarrow E_k(m)$
6: **return** $c = (v, s)$

Algorithm 3 $\texttt{ElGamal.Dec}(\texttt{sk} = \alpha, c = (v, s))$

1: $\hat{w} \leftarrow v^\alpha$
2: $\hat{k} \leftarrow H(\hat{w})$
3: $\hat{m} \leftarrow D_{\hat{k}}(s)$
4: **return** $\hat{m}$

Under the random oracle model, if (G, g) is such that the decisional Diffie-Hellman problem is easy (in other words, there exists efficient algorithm $\mathcal{O}^{\texttt{DDH}}$ for distinguishing Diffie-Hellman triple), then a computational Diffie-Hellman adversary B can simulate decryption oracle for a CCA adversary A against `ElGamal`: when presented with query $(\tilde{v}, \tilde{s})$, B searches through the tape of the hash function $H_{\mathcal{K}}$ for input $\tilde{w}$ such that $(\texttt{pk}, \tilde{v}, \tilde{w})$ is a Diffie-Hellman triple. If no such input exists, then B samples a random key. Because DDH is easy, B can detect when A has queried the CDH answer, at which point B can terminate A and win the CDH game.

Theorem 1. *Under ROM, if $G = \langle g \rangle$ is such that CDH is hard and DDH is easy, then* `ElGamal` *is CCA secure*

We can apply Shamir's secret sharing scheme to convert this into a threshold scheme, which we will denote by `TDH0a`. t, n are the threshold parameters: there are n decryption servers and t or more are needed to decrypt a ciphertext.

Algorithm 4 $\texttt{TDH0a.KeyGen}(t, n)$

1: $\alpha_0, \ldots, \alpha_{t-1} \leftarrow \mathbb{Z}_q$
2: $f(x) \leftarrow \alpha_{t-1}x^{t-1} + \ldots + \alpha_1 x + \alpha_0$
3: **for** $i \in \{1, 2, \ldots, n\}$ **do**
4: $\quad \texttt{sk}_i \leftarrow f(i)$
5: **end for**
6: $\texttt{pk} \leftarrow g^{f(0)}$ (equal to g^{α_0})
7: **return** $(\texttt{pk}, \{\texttt{sk}_i\}_{i=1}^n)$