On a Traitor Tracing Scheme from ACISP 2003

Dongvu Tonien
dong@uow.edu.au

Abstract

At ACISP 2003 conference, Narayanan, Rangan and Kim proposed a secret-key traitor tracing scheme used for pay TV system. In this note, we point out a flaw in their scheme.

1 The Narayanan-Rangan-Kim scheme

Let m be the number of services (data providers), n be the number of users, t be the collusion threshold, and δ be the tolerance bound on accusing innocent users as traitors. Let e denote the Euler constant. The following describes main algorithms in the Narayanan-Rangan-Kim pay TV scheme.

Algorithm Setup: with security parameter 1^ℓ, the setup algorithm does the following.

1. Choose two large primes p, q and set $N = pq$ such that N has ℓ bits;
2. Choose a random number R such that $R\phi(N) + 1$ has a divisor d of roughly ℓ bits;
3. Choose 2ℓ-bit numbers d_1, d_2, d_3 which are divisible by d and $\gcd(d_1, d_3) = d$;
4. Choose random numbers $d_4, d_5, \ldots, d_{t+4} \in \{1, 2, \ldots, \phi(N)\}$;
5. Runs the constraint generation algorithm:
   - Generate $et \log \frac{n}{\delta}$ constraints divided into $h = e \log \frac{n}{\delta}$ groups. A constraint $\gamma = (\mu_0, \mu_1, \mu_2, \ldots, \mu_t, P)$ represents the equation $\sum_{i=0}^{t} \mu_i x_i = 0 \pmod{P}$ where P is a prime. Each constraint group contains t constraints of the same prime;
   - For each $j = 1, \ldots, n$, generate a vector $x = (x_0, x_1, \ldots, x_t) = (e_{4,j}, e_{5,j}, \ldots, e_{t+4,j})$ as follows: select each of the constraints with probability $1 - \frac{1}{t}$; x is constructed so that it satisfies all the selected constraints.

Algorithm AddUser: if a user U_j $(1 \le j \le n)$ joins the system, do the following.

1. Select a random even number $e_{1,j}$;
2. Retrieve vector $(e_{4,j}, e_{5,j}, \ldots, e_{t+4,j})$ from the Setup algorithm;
3. Choose $e_{2,j}$ and $e_{3,j}$ so that $\sum_{r=1}^{t+4} e_{r,j} d_r = R\phi(N) + 1$;
4. Give user U_j the following $(t+4)$-tuple $(e_{1,j}, e_{2,j}, e_{3,j}, e_{4,j}, e_{5,j}, \ldots, e_{t+4,j})$ as his/her secret decryption key.

Algorithm AddStream: if a data provider (or stream) S_i joins the system, do the following.

1. Give $t+4$ secret numbers d_1, d_2, ..., d_{t+4} to S_i;
2. Choose a random $g_i \in Z_N^*$ of high order modulo N;
3. Give S_i the value g_i as its secret encryption key.

Algorithm Subscribe: if a user U_j subscribes to a stream S_i, do the following.

1. Set the subscribe matrix entry $Subsc[i, j] = 1$;
2. Give user U_j the value $g_i^{e_{1,j}}$.

Algorithm Unsubscribe: if a user U_j unsubscribes to a stream S_i, do the following.

1. Set the subscribe matrix entry $Subsc[i, j] = 0$;
2. Reset the value g_i of the stream S_i to a new value $\mathsf{new}\, g_i$;
3. Re-subscribes all users who are currently subscribing to S_i (that is, give each user U_k that subscribes to S_i the new value $\mathsf{new}\, g_i^{e_{1,k}}$).

Algorithm Broadcast: if a stream S_i wants to broadcast a program M, then S_i uses its secret encryption key g_i to do the following.

1. Choose a random number z coprime to $\phi(N)$;
2. Calculate and broadcast the following ciphertext

$$(z, C_1, C_2, C_3, \ldots, C_{t+4}) = (z, M^{d_1} g_i^z, M^{d_2}, M^{d_3}, \ldots, M^{d_{t+4}}).$$

Algorithm Decryption: if user U_j subscribes stream S_i, then U_j can use its secret encryption key $(e_{1,j}, e_{2,j}, \ldots, e_{t+4,j})$ and the value $g_i^{e_{1,j}}$ to decrypt a ciphertext $(z, C_1, C_2, C_3, \ldots, C_{t+4})$ broadcasted by S_i as follows

$$\frac{C_1^{e_{1,j}} C_2^{e_{2,j}} C_3^{e_{3,j}} \ldots C_{t+4}^{e_{t+4,j}}}{(g_i^{e_{1,j}})^z} = M.$$

2 A Flaw

This flaw is in the algorithm AddUser. In the step 3 of this algorithm, two numbers $e_{2,j}$, $e_{3,j}$ must be chosen so that

$$e_{1,j}d_1 + e_{2,j}d_2 + e_{3,j}d_3 + e_{4,j}d_4 + e_{5,j}d_5 + \ldots + e_{t+4,j}d_{t+4} = R\phi(N) + 1.$$

Since d_1, d_2 and d_3 are all divisible by d, the *necessary* condition for this equation is solvable for $e_{2,j}$, $e_{3,j}$ is

$$\Delta_j = e_{4,j}d_4 + e_{5,j}d_5 + \ldots + e_{t+4,j}d_{t+4} - (R\phi(N) + 1) = 0 \pmod{d}.$$

Therefore, we have n equations on $t+1$ numbers d_4, d_5, $\ldots$, d_{t+4} as follows

$$\Delta_1 = e_{4,1}d_4 + e_{5,1}d_5 + \ldots + e_{t+4,1}d_{t+4} - (R\phi(M)+1) = 0 \pmod{d}$$

$$\Delta_2 = e_{4,2}d_4 + e_{5,2}d_5 + \ldots + e_{t+4,2}d_{t+4} - (R\phi(M)+1) = 0 \pmod{d}$$

$$\ldots$$

$$\Delta_n = e_{4,n}d_4 + e_{5,n}d_5 + \ldots + e_{t+4,n}d_{t+4} - (R\phi(M)+1) = 0 \pmod{d}$$

Since n is much larger than t, this is unlikely to be satisfied. Note that in the algorithm Setup, $t+1$ numbers d_4, d_5, $\ldots$, d_{t+4} are randomly chosen independently with the generation of the n vectors $(e_{4,1}, \ldots, e_{t+4,1})$, $(e_{4,2}, \ldots, e_{t+4,2})$, $\ldots$, $(e_{4,n}, \ldots, e_{t+4,n})$.

Since the flaw is in a crucial component, the AddUser algorithm of the system, the pay TV scheme proposed by Narayanan, Rangan and Kim is unusable.

References

[1] A. Narayanan, C.P. Rangan and K. Kim, *Practical Pay TV Schemes,* ACISP'03, LNCS **2727** (2003), pp. 192–203.