# Cryptanalysis of YCN Key Assignment Scheme in a Hierarchy *

Min-Shiang Hwang

Department of Information Management
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, TAIWAN 413 , R.O.C.
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-3742337

March 10, 2009

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC89-2213-E-324-001.

**Abstract**

In this article, we present counter-evidence to point out that the YCN cryptographic key assignment scheme in a hierarchy is not secure.

**Indexing terms:** Cryptography, Data Security, User Hierarchy.

# 1 Introduction

In [9], the authors proposed an efficient cryptographic key assignment scheme (named, the YCN scheme) for solving the access control problem in a hierarchy [1, 2, 4, 5, 6, 7, 8]. An access control policy in a user hierarchy allocates users into security classes $(C_1, C_2, \cdots)$ according to their significance. The security classes form a partially ordered hierarchy, such that a higher security class can access the same or lower security classes, but the opposite is not allowed.

The YCN scheme enforces access control policies in a user matrix model, which is more flexible than that in a user hierarchy. The user matrix model not only can model the access control policies in the user hierarchy model, but also more complicated policies with anti-symmetrical and transitive exceptions. The transitive exception policy is that $C_1$ can access $C_2$ and $C_2$ can access $C_3$, but $C_1$ can not access $C_3$. The anti-symmetrical exception policy is that $C_1$ can access $C_2$ and $C_2$ can access $C_1$, but $C_1$ and $C_2$ are two distinct user classes.

In [10], the authors demonstrated that it is impossible for illegal users to derive the derivation and encryption keys under the YCN scheme. The YCN scheme is simple and efficient in generating and deriving keys. However, we provide counter-evidence here to point out that the YCN cryptographic key assignment scheme in a hierarchy is not secure.

## 2 A Review of the YCN Scheme

In [9], the authors assumed that there is a central authority (CA) in the system that is responsible for generating and distributing keys. The CA assigns each user class, $C_i$, two keys: a derivation and an encryption key. The encryption key is used to encrypt data to protect against illegal access from other user classes. The derivation key is used to derive the encryption keys of other user classes in order to decrypt and access their data. CA generates a secret number $K_0$, and computes the product $M$ of two large prime numbers and makes it public. We briefly review the YCN scheme in the following.

1. Assign a distinct prime number $P_i$ to each user class $C_i$. Suppose $C_i$ has $U_i$ successors $C_{ij}$ $(j = 1, 2, \cdots, U_i)$ excludes immediate successors, CA assigns $U_i$ distinct prime numbers $P_{ij}$ to $C_i$.

2. Compute the product $X_i$ for $C_i$.

$$X_i = \prod_{j=1,\cdots,U_i} P_{ij} \prod_{C_i<C_m, j=1,\cdots,U_m} P_{mj}. \tag{1}$$

   Here, $C_i < C_m$ means that the user class $C_i$ is a successor class of $C_m$.

3. Compute the public information $T_{ie}$ and $T_{id}$ for $C_i$.

$$T_{id} = \prod_{j=1,\cdots,U_i} P_{ij} \prod_{C_m \not\leq C_i} P_m, \tag{2}$$

$$T_{ie} = (X_i / \prod_{C_n} P_{ni}) \prod_{C_m \not\leq C_i} P_m. \tag{3}$$

   Here, $C_n$ is an ancestor class (the immediate ancestor class is excluded) of $C_i$.

4. Assign the derivation key $K_{id}$ and encryption key $K_{ie}$ for each $C_i$.

$$K_{id} = (K_0)^{T_{id}} \bmod M, \tag{4}$$

$$K_{ie} = (K_0)^{T_{ie}} \bmod M. \tag{5}$$

   $K_{id}$ and $K_{ie}$ are kept secret by the user class $C_i$.

The class $C_i$ can use its own derivation key, $K_{id}$, to derive the encryption key, $K_{je}$, of $C_j$ as follows.

$$K_{je} = (K_{id})^{T_{je}/T_{id}} \bmod M. \tag{6}$$

We use the same example in [9] to illustrate the YCN scheme. For the access control policy in a 6-class (as shown in Figure 1) with explicit transitive exception information, Table 1 illustrates the prime numbers and public information. The key assignments using the YCN scheme are shown in Table 2. In the example, $C_1$ can use its own derivation key, $K_{1d}$, to derive the encryption key $K_{2e}$ of $C_2$ and $C_2$ can use its own derivation key, $K_{2d}$, to derive the encryption key $K_{3e}$ of $C_3$. But $C_1$ can not use its own derivation key $K_{1d}$ to derive the encryption key $K_{3e}$ of $C_3$ using the transitive exception policy.

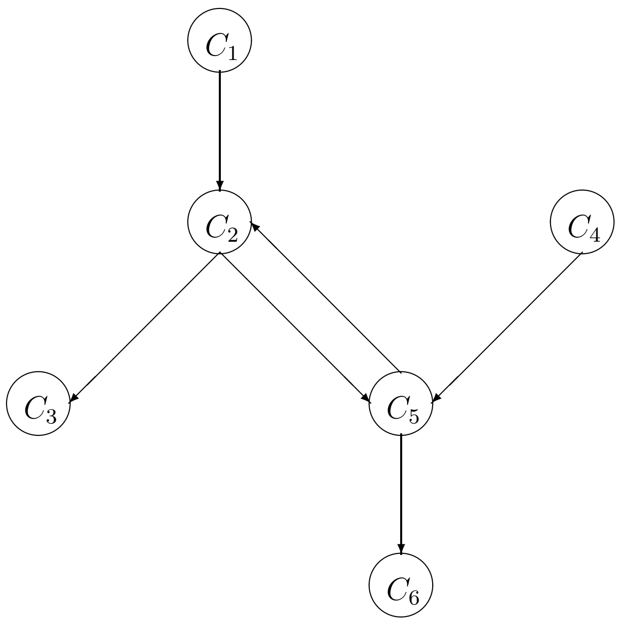


Figure 1: An example of a policy in a hierarchical structure with explicit transitive exception information.

The authors claim that illegal users cannot derive the derivation and encryption keys of other user classes. However, we show that several user classes can collaborate to derive the keys of other user classes in the next session.

Table 1: The prime numbers and public information of each user class.

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
|---|---|---|---|---|---|---|
| $P_i$ | 2 | 3 | 5 | 7 | 11 | 13 |
| $P_{ij}$ | $P_{13} = 17$<br>$P_{15} = 19$<br>$P_{16} = 23$ | $P_{26} = 29$ | | $P_{42} = 31$<br>$P_{43} = 37$<br>$P_{46} = 41$ | $P_{53} = 43$ | |
| $X_i$ | $17 \cdot 19 \cdot 23$ | $17 \cdot 19 \cdot 23$<br>$29 \cdot 31 \cdot 37$<br>$41 \cdot 43$ | $17 \cdot 19 \cdot 23$<br>$29 \cdot 31 \cdot 37$<br>$41 \cdot 43$ | $31 \cdot 37 \cdot 41$ | $17 \cdot 19 \cdot 23$<br>$29 \cdot 31 \cdot 37$<br>$41 \cdot 43$ | $17 \cdot 19 \cdot 23$<br>$29 \cdot 31 \cdot 37$<br>$41 \cdot 43$ |
| $T_{id}$ | $7 \cdot 17 \cdot 19$<br>$23$ | $2 \cdot 7 \cdot 29$ | $2 \cdot 3 \cdot 7$<br>$11 \cdot 13$ | $2 \cdot 31 \cdot 37$<br>$41$ | $2 \cdot 7 \cdot 43$ | $2 \cdot 3 \cdot 5$<br>$7 \cdot 11$ |
| $T_{ie}$ | $7 \cdot 17 \cdot 19$<br>$23$ | $2 \cdot 7 \cdot 17$<br>$19 \cdot 23 \cdot 29$<br>$37 \cdot 41 \cdot 43$ | $2 \cdot 3 \cdot 7$<br>$11 \cdot 13 \cdot 19$<br>$23 \cdot 29 \cdot 31$<br>$41$ | $2 \cdot 31 \cdot 37$<br>$41$ | $2 \cdot 7 \cdot 17$<br>$23 \cdot 29 \cdot 31$<br>$37 \cdot 41 \cdot 43$ | $2 \cdot 3 \cdot 5$<br>$7 \cdot 11 \cdot 17$<br>$19 \cdot 31 \cdot 37$<br>$43$ |

Table 2: The derivation and encryption keys held by each user class.

| | $K_{id}$ | $K_{ie}$ |
|---|---|---|
| $C_1$ | $K_{1d} = K_0^{7 \cdot 17 \cdot 19 \cdot 23} \bmod M$ | $K_{1e} = K_0^{7 \cdot 17 \cdot 19 \cdot 23} \bmod M$ |
| $C_2$ | $K_{2d} = K_0^{2 \cdot 7 \cdot 29} \bmod M$ | $K_{2e} = K_0^{2 \cdot 7 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43} \bmod M$ |
| $C_3$ | $K_{3d} = K_0^{2 \cdot 3 \cdot 7 \cdot 11 \cdot 13} \bmod M$ | $K_{3e} = K_0^{2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 41} \bmod M$ |
| $C_4$ | $K_{4d} = K_0^{2 \cdot 31 \cdot 37 \cdot 41} \bmod M$ | $K_{4e} = K_0^{2 \cdot 31 \cdot 37 \cdot 41} \bmod M$ |
| $C_5$ | $K_{5d} = K_0^{2 \cdot 7 \cdot 43} \bmod M$ | $K_{5e} = K_0^{2 \cdot 7 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43} \bmod M$ |
| $C_6$ | $K_{6d} = K_0^{2 \cdot 3 \cdot 5 \cdot 7 \cdot 11} \bmod M$ | $K_{6e} = K_0^{2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 43} \bmod M$ |

# 3 The Weakness of the YCN Scheme

In this section, we present some cases to show that the YCN scheme is not secure. The weakness in the security of the YCN scheme is shown in the following theorems. We assume that there are $m$ user classes $(C_i, i = 1, 2, \cdots, m)$ in a hierarchy structure. When we say $C_i$ is the top class in the hierarchy. This means is that $C_i$ does not have an ancestor class.

**Theorem 3.1** *Assume that there are only two top classes ($C_a$ and $C_b$) in the hierarchy. $C_a$ and $C_b$ can collaborate to derive the derivation and encryption keys of all of the classes in the YCN scheme.*

**Proof.** The equations for $T_{ad}$ and $T_{bd}$ in Equation 2 are as follows.

$$\begin{aligned} T_{ad} &= \prod_{j=1,\cdots,U_a} P_{aj} \prod_{C_m \not\leq C_a} P_m, \\ T_{bd} &= \prod_{j=1,\cdots,U_b} P_{bj} \prod_{C_m \not\leq C_b} P_m. \end{aligned}$$

Because $\prod_{j=1,\cdots,U_a} P_{aj}$ and $\prod_{j=1,\cdots,U_b} P_{bj}$ are distinct product of primes and no common factor, $\prod_{C_m \not\leq C_a} P_m = P_b$, and $\prod_{C_m \not\leq C_b} P_m = P_a$. Therefore, $\gcd(T_{ad}, T_{bd}) = 1$, that is $T_{ad}$ and $T_{bd}$ are relatively prime. There thus exists two integers $s$ and $t$ such that $sT_{ad} + tT_{bd} = 1$ [3]. $C_a$ and $C_b$ can collaborate to derive the secret $K_0$ using their derivation keys as follows.

$$\begin{aligned} K_{ad}^s K_{bd}^t &= (K_0)^{sT_{ad}} (K_0)^{tT_{bd}} \bmod M \\ &= (K_0)^{(sT_{ad}+tT_{bd})} \bmod M \\ &= K_0. \end{aligned}$$

Since all $T_{id}$ and $M$ are public, the derivation key $K_{id}$ of all classes can be computed as in Equation 4. In additions, each class can use it's own derivation key $K_{id}$ to derive its own encryption key $K_{ie}$. The proof thus holds. Q.E.D.

**Example 3.1** *We use the same example as in Figure 1, Table 1, and Table 2. Since $C_1$ and $C_4$ are the top two user classes in the hierarchy,* $\gcd(T_{1d}, T_{4d}) =$

$\gcd(52003, 94054) = 1$ *and exists two integers* $(s, t) = (76107, -42080)$ *such that* $sT_{1d} + tT_{4d} = 1$. *Therefore,* $C_1$ *and* $C_4$ *can collaborate to derive the secret* $K_0$ *using their derivation keys* $(K_{1d}, K_{4d})$ *as follows.*

$$\begin{aligned} K_{1d}^s K_{4d}^t &= (K_0)^{sT_{1d}} (K_0)^{tT_{4d}} \bmod M \\ &= (K_0)^{((76107 \times 52003) - (42080 \times 94054))} \bmod M \\ &= K_0. \end{aligned}$$

*The decryption key of* $C_3$ *thus can be derived as follows.*

$$\begin{aligned} & (K_{1d}^s K_{4d}^t)^{T_{3d}} \bmod M \\ = \;& (K_0)^{T_{3d}} \bmod M \\ = \;& K_{3d}. \end{aligned}$$

*The encryption key of* $C_6$ *also can be derived as follows.*

$$\begin{aligned} & (K_{1d}^s K_{4d}^t)^{T_{6e}} \bmod M \\ = \;& (K_0)^{T_{6e}} \bmod M \\ = \;& K_{6e}. \end{aligned}$$

*Notes that* $C_1$ *and* $C_4$ *can not access* $C_3$ *and* $C_6$ *in the hierarchy using the transitive exception policy.*

**Theorem 3.2** *If* $C_1$, $C_2$, $\cdots$,*and* $C_n$ *are* $n$ *top classes in the hierarchy, any two of these classes (e.g.* $C_1$ *and* $C_2$*) can collaborate to derive the derivation and encryption keys of all successors of these top classes.*

**Proof.** The equations for $T_{1d}$ and $T_{2d}$ in Equation 2 are as follows.

$$\begin{aligned} T_{1d} &= \prod_{j=1,\cdots,U_1} P_{1j} \prod_{C_m \not\leq C_1} P_m, \\ T_{2d} &= \prod_{j=1,\cdots,U_2} P_{2j} \prod_{C_m \not\leq C_2} P_m. \end{aligned}$$

Since $\prod_{j=1,\cdots,U_1} P_{1j}$ and $\prod_{j=1,\cdots,U_2} P_{2j}$ are distinct prime numbers, $\prod_{C_m \not\leq C_1} P_m = P_2 P_3 \cdots P_n$, and $\prod_{C_m \not\leq C_2} P_m = P_1 P_3 \cdots P_n$. Therefore, $\gcd(T_{1d}, T_{2d}) = P_3 P_4 \cdots P_n$.

Let $Y_1 = P_2 \prod_{j=1,\cdots,U_1} P_{1j}$ and $Y_2 = P_1 \prod_{j=1,\cdots,U_2} P_{2j}$. We can produce $\gcd(Y_1, Y_2) = 1$. There thus exists two integers $s$ and $t$ such that $sY_1 + tY_2 = 1$. Therefore, $C_1$ and $C_2$ can collaborate to derive $(K_0)^{P_3 P_4 \cdots P_n}$ using their derivation keys as follows.

$$\begin{aligned} K_{1d}^s K_{2d}^t &= (K_0)^{sT_{1d}} (K_0)^{tT_{2d}} \bmod M \\ &= (K_0)^{(sT_{1d}+tT_{2d})} \bmod M \\ &= (K_0)^{P_3 P_4 \cdots P_n} \bmod M. \end{aligned}$$

Since the great common divisor of all successors of $C_1$, $C_2$, $\cdots$, and $C_n$ is $(K_0)^{P_3 P_4 \cdots P_n}$, $C_1$ and $C_2$ can easily derive the derivation and encryption keys $K_{id}$ and $K_{ie}$ of these successors. The proof thus holds. Q.E.D.

Assume that there are $n$ top classes in the hierarchy, one of these top classes (e.g. $C_1$) and one of its successors can also collaborate to derive the derivation and encryption keys of all successors of $C_1$. For example, $C_1$ and $C_2$ in Figure 1 can derive the derivation and encryption keys of $C_6$. Since $\gcd(T_{1d}, T_{2d}) = 7$, there exists $s$ and $t$ such that $s(T_{1d}/7) + t(T_{2d}/7) = 1$. $C_1$ and $C_2$ can collaborate to derive the secret $(K_0)^7$ using their derivation keys $(K_{1d}, K_{2d})$. Therefore, $K_{6d}$ can be derived as follows.

$$\begin{aligned} & ((K_{1d})^s (K_{2d})^t)^{T_{6d}/7} \bmod M \\ = \ & ((K_0)^{sT_{1d}} (K_0)^{tT_{2d}})^{T_{6d}/7} \bmod M \\ = \ & (K_0)^{T_{6d}} \bmod M \\ = \ & K_{6d}. \end{aligned}$$

In fact, any two user classes in a hierarchy, in which one member of that class is an ancestor class of the other class, can collaborate to derive the derivation and encryption keys of all successors of the two classes. For example, $C_5$ and $C_6$ in Figure 1 can derive the derivation and encryption keys of $C_3$. Since $\gcd(T_{5d}, T_{6d}) = 2 \times 7 = 14$, there exists $s$ and $t$ such that

$s(T_{5d}/14) + t(T_{6d}/14) = 1$. $C_5$ and $C_6$ thus can collaborate to derive the secret $(K_0)^{14}$ using their derivation keys $(K_{5d}, K_{6d})$. Therefore, $K_{3d}$ can be derived as follows.

$$\begin{aligned}
& ((K_{5d})^s (K_{6d})^t)^{T_{3d}/14} \bmod M \\
= \; & ((K_0)^{sT_{5d}} (K_0)^{tT_{6d}})^{T_{3d}/14} \bmod M \\
= \; & (K_0)^{T_{3d}} \bmod M \\
= \; & K_{3d}.
\end{aligned}$$

# 4 Conclusion

We have shown how several user classes in the YCN scheme can collaborate to derive the derivation and encryption keys of other user classes in some cases. Although the YCN scheme is not secure, it has opened a brand new research area for key assignment scheme in a hierarchy.

# References


[1] S.G. AKL and P.D. TAYLOR, 'Cryptographic solution to a problem of access control in a hierarchy', *ACM Transactions on Computer Systems,* vol. 1, pp. 239–248, 1983.

[2] G.C. Chick and S.E. Tavares, 'Flexible access control with master keys', in proceedings *Advances in Cryptology, CRYPTO'89,* 1990, pp. 316-322.

[3] D.E.R. Denning, 'Cryptography and Data Security', Addison-Wesley, pp.35-54, 1982.

[4] M.S. Hwang, 'A cryptolographic key assignment scheme in a hierarchy for access control', *Mathematical and Computer Modelling*, vol. 26, no. 2, pp. 27–31, 1997.

[5] M.S. Hwang, 'An improvement of a dynamic cryptographic key assignment scheme in a tree hierarchy', *Computers & Mathematics with Applications*, vol.37, no.3, pp.19-22, 1999.

[6] M.S. Hwang, 'An improvement of novel cryptographic key assignment scheme for dynamic access control in a hierarchy', *IEICE Transactions on Fundamentals*, vol. E82-A(3), pp. 548–550, Mar. 1999.

[7] M.S. Hwang, 'Extension of CHW cryptographic key assignment scheme in a hierarchy', accepted and to appear in *IEE Proceedings Computers and Digital Techniques*, 1999.

[8] R.S. Sandhu, 'Cryptographic implementation of a tree hierarchy for access control', *Information Processing Letters*, vol. 27, pp. 95–98, 1988.

[9] J.H. Yeh, R. Chow, and R. Newman, 'A key assignment for enforcing access control policy exceptions', in proceedings *The International Symposium on Internet Technology,* May, 1998, Taipei, pp. 54-59.

[10] J.H. Yeh, R. Chow, and R. Newman, 'A cryptographic key assignment for access control in a user matrix model', Univ. of Florida, CISE Department, Technical Report, 1998.