Attacks on a Secure Group Communication Scheme With Hierarchical Access Control

Willi Geiselmann and Rainer Steinwandt

IAKS, Arbeitsgruppe Systemsicherheit, Prof. Beth,
Fakultät für Informatik, Universität Karlsruhe,
Am Fasanengarten 5, 76 131 Karlsruhe, Germany

Abstract. At ICICS 2001, Zou, Ramamurthy, and Magliveras proposed CRTHACS, a chinese remainder theorem based scheme for secure group communication with hierarchical access control. The scheme is designed in such a way that the underlying hierarchy remains hidden from the participating parties/users. This contribution describes several practical attacks on CRTHACS which can reveal significant parts of the hierarchy.

Keywords: hierarchical access control, cryptanalysis

1 Introduction

In a scenario where the set of parties/users is divided into subgroups with different privilege levels, the problem of secure group communication with hierarchical access control arises. The basic idea is to enable each subgroup to receive and decrypt the messages of those subgroups that are located at a lower level in the hierarchy. To cope with such a setting, in [1] Zou, Ramamurthy, and Magliveras propose a scheme called CRTHACS, which uses a combination of several cryptographic primitives to achieve the desired security guarantees. CRTHACS is a so-called *independent key scheme* where the encryption keys of the individual subgroups can be chosen independently, and the scheme is designed to also hide the underlying hierarchy.

After recalling the basic set-up of CRTHACS in the next section, we show that in the proposed form this scheme is vulnerable to some annoyingly simple but practical attacks based on Euclid's algorithm. Namely, it is possible to reveal at least parts of the hierarchy after eavesdropping sufficiently many transmissions; once each subgroup has sent about half a dozen of messages, the attack has already good chances to succeed. Hereafter, in Section 4, we show that a collusion of at least two malicious subgroups is able to reveal common ancestors in the hierarchy by means of their local data alone, i. e., without having to eavesdrop transmissions. Finally, we show how a single subgroup can use its local information for revealing its ancestors in the hierarchy. Thus, in the proposed form CRTHACS does not offer the security level originally aimed at.

2 Description of CRTHACS

Initially, the group of users is split into disjoint subgroups $G_1,\ldots,G_m$. Moreover, we are given a hierarchy among the subgroups, which can be specified through a directed acyclic graph: A directed path from G_i to G_j means that G_i is located above G_j in the hierarchy.

Inside each subgroup G_i a distinguished party acts as *subgroup controller*. This party is responsible for the key management inside its subgroup, and for our purposes we can identify the subgroup controllers with their respective subgroup. Further on, the Chinese Remainder Theorem Based Hierarchical Access Control Scheme (CRTHACS) from [1] makes use of a trusted party GC which can be located outside the hierarchy. GC acts as *group controller*, i. e., it is in charge of adding, inserting, and removing subgroups. GC is also involved in the initialization phase where all subgroups/subgroup controllers are equipped with the key material needed for the subsequent secure group communication with hierarchical access control. For details on this initialization phase we refer to the original paper [1]. Here it is sufficent to know that after some initial communication, each subgroup G_i possesses a six-tuple $(P_i, S_i, K_i, N_i, COM\_CRT_i, \mathcal{N}_i)$, such that for all $i \in \{1, \ldots, m\}$

- (P_i, S_i) is a (public key, secret key) pair generated by the subgroup controller G_i;
- K_i is a key for some symmetric cipher, that is chosen and used by the subgroup G_i to encrypt data;
- N_i, COM_CRT_i, and $\mathcal{N}_i$ are integer values obtained from the trusted group controller GC. They satisfy the following conditions:
  - $N_1, \ldots, N_m$ are pairwise relatively prime,
  - $\mathcal{N}_i := \prod_{G_j \text{ is an ancestor of } G_i} N_j$,
  - $COM\_CRT_i \equiv E_{P_j}(K_i) \pmod{N_j}$ for all ancestors G_j of G_i; here $E_{P_j}(\cdot)$ denotes (public key) encryption with P_j. Motivated by the varying use of secret and public keys as arguments for $E_{(\cdot)}(\cdot)$ in [1], we assume that $E_{P_j}(\cdot)$ is a function, i. e., that encryption is not probabilistic.

Finally, there is another publicly known integer, N_0, which is coprime to $N_1, \ldots, N_m$; this number is needed for transmitting ciphertexts within the hierarchy. Note here that only $P_1, \ldots, P_m$, and N_0 are made public, all other values remain secret. In particular, a subgroup is not supposed to know its ancestors. The hierarchy is known to the trusted GC, but supposed to remain hidden from the subgroups G_i.

Once the initialization phase of CRTHACS is complete, a member of a group G_i with identity ID_j proceeds as follows to send a message M:

1. M is encrypted with K_i. Let $\{M\}_{K_i}$ be the resulting ciphertext.
2. Next, a keyed MAC of $\{M\}_{K_i}$ is computed; as in [1] we denote this value by $MAC_{K_i}(\{M\}_{K_i})$.
3. Finally, a solution CRT_i of the pair of congruences

$$\begin{aligned} CRT_i &\equiv COM\_CRT_i \pmod{\mathcal{N}_i} \\ CRT_i &\equiv E_{S_j}(MAC_{K_i}(\{M\}_{K_i})) \pmod{N_0} \end{aligned} \tag{1}$$

   is computed.
4. The triple $(ID_j, CRT_i, \{M\}_{K_i})$ is sent via a broadcast or multicast.

The details of how a message is legitimately received and decrypted are not relevant for our attacks, and we refer to [1] for details on this issue. The basic idea is that by construction any ancestor of the subgroup G_i can recover K_i from CRT_i; the MAC aims at ensuring authenticity and integrity of the message.

3 Eavesdropping messages to learn about the hierarchy

Let us assume that at least three different messages $M_{i,1}$, $M_{i,2}$, $M_{i,3}$ have been sent by (not necessarily distinct) members of a subgroup G_i. We denote the corresponding solutions of the congruence system (1) by $CRT_{i,1}$, $CRT_{i,2}$, and $CRT_{i,3}$. Then we know that $CRT_{i,1} \equiv CRT_{i,2} \equiv CRT_{i,3} \equiv COM\_CRT_i \pmod{\mathcal{N}_i}$, and therefore both

$$\begin{aligned} CRT_{i,1} - CRT_{i,2} &\equiv 0 \pmod{\mathcal{N}_i} \text{ and} \\ CRT_{i,1} - CRT_{i,3} &\equiv 0 \pmod{\mathcal{N}_i} \end{aligned} \tag{2}$$

must hold. To an attacker who eavesdrops the three transmitted values $CRT_{i,1}$, $CRT_{i,2}$, $CRT_{i,3}$, the value $\mathcal{N}_i$ is a priori not known. But as condition (2) is known to be fulfilled, the attacker simply computes $\mathcal{N}_i' := \gcd(CRT_{i,1} - CRT_{i,2}, CRT_{i,1} - CRT_{i,3})$, and with some luck she has $\mathcal{N}_i' = \mathcal{N}_i$. Lacking a concrete specification for choosing the $\mathcal{N}_i$'s, we cannot give a quantitative estimate here. But it is plausible that with, say, a dozen of messages sent from within G_i, the attacker's chances of learning $\mathcal{N}_i$ through simple gcd-computations are extremely good.

Applying this observation to each of the subgroups $G_1,\ldots,G_m$, we see that an attacker has good chances to learn all $\mathcal{N}_i$-values, once each subgroup has submitted a sufficient number of messages (where already three messages per subgroup can suffice). Knowing $\mathcal{N}_1,\ldots,\mathcal{N}_m$, we can derive necessary conditions for the existence of directed paths in the acyclic graph specifying the hierarchy: For G_i being an ancestor of G_j, the condition $\mathcal{N}_i \mid \mathcal{N}_j$ must be fulfilled. In this way, information about the underlying hierarchy can be revealed, e. g., from an eavesdropping "outsider". The next sections show that from within the hierarchy, more powerful attacks are possible.

4 Revealing common ancestors without eavesdropping: colluding subgroups

Assume that two subgroups G_a and G_b want to identify their common ancestors without revealing their private data to each other. To do so, they can proceed as follows: First, they fix an arbitrary message M and use the congruences (1) to derive the corresponding values CRT_a, CRT_b. Then, by definition of COM_CRT_a and COM_CRT_b, we know that for each common ancestor G_c of G_a and G_b, the condition

$$\gcd(CRT_a - E_{P_c}(K_a), CRT_b - E_{P_c}(K_b)) \equiv 0 \pmod{N_c}$$

holds. The key P_c is public, and hence computing this greatest common divisor provides no difficulties to the collaborating subgroups G_a and G_b. Further on, by construction of CRTHACS, it is reasonable to assume that the value N_c is quite large, e. g., when using RSA, a bit length of ≥ 1024 seems plausible. Thus, whenever the greatest common divisor of $CRT_a - E_{P_c}(K_a)$ and $CRT_b - E_{P_c}(K_b)$ has a length of, say, more than a few hundred bits, then G_a and G_b have good reason to believe that G_c is a common ancestor of them. On the other hand, if this greatest common divisor is small, then G_c cannot be a common ancestor of G_a and G_b.

5 Revealing ancestors from local data alone

The observation that the N_i-values are quite large, can also be exploited by an individual subgroup: Let G_a encrypt an arbitrary message and compute a corresponding value CRT_a by means of the congruences (1). Then, in analogy to the attack in the previous section, for any ancestor G_c of G_a we have

$$\gcd(CRT_a - E_{P_c}(K_a), \mathcal{N}_a) \equiv 0 \pmod{N_c}.$$

Consequently, the bitsize of this greatest common divisor offers G_a a good chance to learn whether G_c is one of its ancestors.

6 Conclusion

The above discussion shows that CRTHACS is vulnerable to some annoyingly simple but practical attacks based on Euclid's algorithm. Thus, in the proposed form, CRTHACS does not ensure that the hierarchy remains hidden, as originally intended.

Acknowledgments

We are indebted to the authors of [1] for valuable discussions on their scheme. In particular, the attacks in Section 4 and 5 were inspired by modified versions of CRTHACS that have been put forward in private communication.

References

1. X. Zou, B. Ramamurthy, and S.S. Magliveras. Chinese Remainder Theorem Based Hierarchical Access Control for Secure Group Communication. In S. Qing, T. Okamoto, and J. Zhou, editors, *Information and Communications Security: Third International Conference, ICICS 2001*, volume 2229 of *Lecture Notes in Computer Science*, pages 381–385. Springer, 2001.