Corrections to "ALMS: Asymmetric Lightweight Centralized Group Key Management Protocol for VANETs"

Ahmad Mansour, Khalid M. Malik, *Senior Member, IEEE*, Ahmed Alkaff, and Hisham Kanaan

In the above article [1], on pages 11 and 14, math appears incorrectly in some paragraphs.

On page 11, the math should correctly read as:

To calculate time complexity, we make certain assumptions. First, basic arithmetic operations such as addition and subtraction have time complexity $O(\lfloor log(n) \rfloor + 1)$. Second, multiplication, division, and modulus have time complexity $O((\lfloor log(n) \rfloor + 1)^2)$. Third, for very large numbers, the extended Euclidean algorithm takes $O(\lfloor log(n) \rfloor^2)$, where n is the largest operand [32].

Based on these assumptions, initialization of *ALMS* takes $O(m \cdot (6 \cdot (\lfloor log(X) \rfloor + 1)^2 + \lfloor log(X) \rfloor^2 + 2 \cdot (\lfloor log(N_i) \rfloor + 1)))$, where X is the largest operand and m is the number of registered vehicles. *ALMS*'s initialization requires *two* multiplications and *one* addition to compute N_i', *one* multiplication and *one* addition to compute e_i', *one* multiplication of N_i to get W, performing the extended Euclidean algorithm to find A_i, and *two* multiplications to get S_i. Now as X grows, the cost of other factors becomes negligible. Therefore, time complexity of initialization is $O(m \cdot ((\lfloor log(X) \rfloor + 1)^2 + \lfloor log(X) \rfloor^2))$. Group formation of *ALMS* takes $O((2 \cdot n + 1) \cdot (\lfloor log(X) \rfloor + 1)^2)$, where n is the number of vehicles in the receiving group. Moreover, it requires n multiplications to get X, and n multiplications with *one* modulus to compute S_{group}. Now as n grows, the constants become negligible. Therefore, the time complexity of group formation is $O(n \cdot (\lfloor log(X) \rfloor + 1)^2)$. Group key computation of *ALMS* takes $O(2 \cdot (\lfloor log(S_{group}) \rfloor + 1)^2)$, where S_{group} is the group encryption parameter that is computed to encrypt the group key, as shown in equation (13). We multiply sk by S_{group}, then take *mod* X to get gk, as shown in equation (15). Now as S_{group} grows, the constant becomes negligible. Therefore, the time complexity of group key computation is $O((\lfloor log(S_{group}) \rfloor + 1)^2)$. Group key retrieval of *ALMS* takes $O(3 \cdot (\lfloor log(gk) \rfloor + 1)^2)$ for one recipient, where gk is the encrypted group key. We take *mod* k_i, then multiply the result by the random number y_i, and we take *mod* a_i to get sk, as shown in equation (16). Now as gk grows, the constant becomes negligible. Therefore, the time complexity of group key retrieval is $O((\lfloor log(gk) \rfloor + 1)^2)$.

Furthermore, the change in group membership takes $O(n \cdot (\lfloor log(S_{group}) \rfloor + 1)^2)$ when a new member joins the group and $O((\lfloor log(S_{group}) \rfloor + 1)^2)$ when a member leaves the group, where n is the number of vehicles in the receiving group. For group member addition, it requires *one* multiplication to get X', n multiplications with *one* modulus to compute S_{group}, and *one* multiplication with *one* modulus to get gk'. Therefore, *ALMS* has a time complexity of $O((n + 4) \cdot (\lfloor log(S_{group}) \rfloor + 1)^2)$. Now as S_{group} grows, the constant becomes negligible. Therefore, the time complexity of new member addition is $O(n \cdot (\lfloor log(S_{group}) \rfloor + 1)^2)$. Group key computation, after an existing member leaves the group, takes *one* division to get X', *one* subtraction with *two* modulus operations to get S_{group}', and *one* multiplication with *one* modulus to get gk'. Therefore, *ALMS* has a time complexity of $O(5 \cdot (\lfloor log(S_{group}) \rfloor + 1)^2 + \lfloor log(S_{group}) \rfloor + 1)$. Now as S_{group} grows, the cost of the addition operation and the constants becomes negligible. Therefore the time complexity of member removal is $O((\lfloor log(S_{group}) \rfloor + 1)^2)$. On the other hand, the storage cost of *ALMS* is $2 \cdot n$ for the TA and *three* for each recipient. The TA needs to store every S_i and N_i, where each recipient needs to store the private key elements k_i, a_i, and y_i. For the communication cost, *ALMS* requires *one broadcast* to share the encrypted group key gk with all receiving group members, after group key computation phase.

On page 14, the math should correctly read as:

To solve this issue, a size reduction of parameters needed to compute the encrypted group key is performed, as shown in section IV-C. It is clear that there is a slight overhead in performing this reduction. However, it will occur only once per receiving group at the TA side, which is negligible considering the gain in terms of computational and communication cost for both the TA and group members. Moreover, when performing the offline registration and initialization, *ALMS* obtains a huge performance gain in the group key computation phase by reducing its complexity from $O(n \cdot ((\lfloor log(X) \rfloor + 1)^2 + \lfloor log(X) \rfloor^2))$ to $O((\lfloor log(X) \rfloor + 1)^2)$, where n is the group size and X is the multiplication of all N_is of group members, as shown in section IV-B. It is important to mention that this huge computational optimization at the TA does not affect the size of the encrypted group key.

Manuscript received October 7, 2020; accepted October 7, 2020. Date of current version March 1, 2021. *(Corresponding author: Khalid M. Malik.)*

Ahmad Mansour is with Dura Automotive Systems LLC, Auburn Hills, MI 48326 USA (e-mail: aamansour@oakland.edu).

Khalid M. Malik is with the Department of Computer Science and Engineering, Oakland University, Rochester, MI 48309 USA (e-mail: mahmood@oakland.edu).

Ahmed Alkaff is with TiraForIT for IT Solutions, Amman 11185, Jordan (e-mail: a.alkaff@tiraforit.com).

Hisham Kanaan is with Secure-24, Southfield, MI 48033 USA (e-mail: hisham.kanaan@secure-24.com).

Digital Object Identifier 10.1109/TITS.2020.3029936

References

[1] A. Mansour, K. M. Malik, A. Alkaff, and H. Kanaan, "ALMS: Asymmetric lightweight centralized group key management protocol for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1663–1678, Mar. 2021, doi: 10.1109/TITS.2020.2975226.