Flaws in Some Self-Healing Key Distribution Schemes with Revocation

Vanesa Daza[1], Javier Herranz [2] and Germán Sáez[2]

[1] Dept. Tecnologies de la Informació i les Comunicacions,
Universitat Pompeu Fabra,
Pg. Circumval·lació 8, Barcelona, Spain
vanesa.daza@upf.cat

[2] Dept. Matemàtica Aplicada IV,
Universitat Politècnica de Catalunya,
C. Jordi Girona 1-3, Barcelona, Spain
{jherranz,german}@ma4.upc.edu

Abstract

Dutta and Mukhopadhyay have recently proposed some very efficient self-healing key distribution schemes with revocation. The parameters of these schemes contradict some results (lower bounds) presented by Blundo et al. In this paper different attacks against the schemes of Dutta and Mukhopadhyay are explained: one of them can be easily avoided with a slight modification in the schemes, but the other one is really serious.

Keywords: self-healing key distribution, lower bounds, polynomial interpolation.

1 Introduction

Self-healing key distribution schemes enable large and dynamic groups of users of an *unreliable* network to establish group keys for secure communication in different sessions. Every session group of users is established by a group manager, by joining or revoking users from the initial group. The common key of the group is provided by the group manager using broadcast packets, which must be combined by the group members with some private information that they have received in the setup phase of the scheme. The goal of this kind of schemes is well captured by the self-healing property: if during a certain session a user loses some broadcast packet, he is still able to recover the group key of the session simply by using a packet received during a previous session and a packet received in a subsequent one, without having to request any additional transmission to the group manager. Thanks to the self-healing property, these key distribution schemes are very useful in several Internet-related and wireless settings.

Different parameters are considered to evaluate the efficiency of self-healing schemes. With respect to the length of the secret information that each user receives from the group manager in the setup phase, Dutta and Mukhopadhyay [1, 2] have recently proposed the most efficient self-healing key distribution schemes up to date. After looking at their proposals, we observed that the schemes were *too* efficient in terms of the length of the secret information of each user, because the achieved efficiency contradicts some theoretical results that have been proved by Blundo et al. [3, 4]. For this reason, we studied the proposals of [1, 2] in more detail and we found some explicit attacks against their security. The conclusion is that the schemes of Dutta and Mukhopadhyay are not secure at all, and that previous proposals already achieved the optimal efficiency in terms of the length of the secret information that each user must store.

Organization. In Section 2 we review the notion of a self-healing key distribution scheme, the required properties and one of the specific proposals of Dutta and Mukhopadhyay (the other ones are very similar). In Section 3 we argue why the security of this proposal would contradict some theoretical results, and then we explain our explicit attacks against the security of the scheme. The conclusions of our work are given in Section 4.

2 Self-Healing Key Distribution with Revocation

Self-healing key distribution schemes were introduced by Staddon et al. [5]. After that, many papers have appeared which generalize and/or modify the original definitions, give lower bounds to the resources required for such schemes, and propose some efficient constructions. See [6, 3, 7, 4] for some relevant papers on self-healing key distribution schemes. Typically a self-healing key distribution scheme consists of the following algorithms.

In the **Setup** phase, the group manager gives to every user U_i in the first session group G_0 his secret information S_i.

In the **Broadcast** phase, for session $j \in \{1, \ldots, m\}$, the group manager makes public some information B_j, which usually depends on the set $R_j \subset G_{j-1}$ of revoked users in session j.

In a **Join** action, the group manager gives to a new user U_i, who joins the group in session j, his secret information S_i which allows him to compute session keys from session j on. The group G_j is formed by the users in G_{j-1} that are not in R_j, along with these new users who have joined the group in session j.

The **Computation of the session key** K_j by a user $U_i \notin R_j$ is performed by using his own secret information S_i and the broadcast packet B_j. The self-healing property of these schemes gives an alternative way to compute the key K_j, by means of the broadcast messages B_{j_1}, B_{j_2}, where $1 \leq j_1 < j < j_2 \leq m$ and $U_i \notin \bigcup_{j=j_1}^{j_2} R_j$.

These schemes usually support *t-revocation* for some threshold t, meaning that the maximum number of users who may have been revoked at some time is t; this is ensured, in particular, if condition $\sum_{j=1}^{m} |R_j| \leq t$ holds. With respect to security, forward and backward secrecy are properties required to a self-healing scheme with t-revocation. *Forward secrecy* means that a set R with $|R| \leq t$ users who have been revoked at session j (or before) cannot obtain any information about the session keys $K_j, K_{j+1}, \ldots, K_m$ from their secret information $\{S_i\}_{U_i \in R}$ and all the broadcast messages $B_1, \ldots, B_m$. *Backward secrecy* means that a set J with $|J| \leq t$ users who join the group after session j cannot obtain any information about the session keys $K_1, \ldots, K_j$ from their secret information $\{S_i\}_{U_i \in J}$ and all the broadcast messages $B_1, \ldots, B_m$.

Most of the existing papers on self-healing key distribution are mainly focused on unconditionally secure schemes: the schemes must achieve forward and backward secrecy even under the attack of an adversary with unlimited computational resources.

2.1 A Scheme of Dutta and Mukhopadhyay

In [1, 2], Dutta and Mukhopadhyay have proposed the most efficient self-healing key distribution schemes with revocation up to date, with respect to the length of the secret information S_i that users must store. We quickly review the protocols of one of these schemes, in [1] (the schemes in [2] are very similar). For simplicity of notation, a user who is revoked at some session j is assumed to remain revoked for the following sessions; in other words, $R_j \subset R_{j+1}$, for $j = 1, \ldots, m-1$.

Setup. The group manager, GM, chooses a finite field F_q (for some prime number q) and a random bivariate t-degree polynomial $\Psi(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + \ldots + a_{t,t}x^t y^t$ in $F_q[x, y]$. This polynomial $\Psi(x, y)$ is kept secret by GM. GM also chooses a one-way permutation $f : F_q \to F_q$, which is made public, and an initial value $\alpha_0 \in F_q$. Each user $U_i \in G_0$ receives as secret information the value α_0 and the polynomial $S_i(y) = \Psi(i, y)$ (i.e., the total information that each user must secretly store consists of $(t+2) \log q$ bits). The group manager finally chooses at random a secret value $K_0 \in F_q$.

Broadcast. In the j-th session, for $j = 1, \ldots, m$, the group manager computes $\alpha_j = f(\alpha_{j-1})$, stores α_j and erases α_{j-1}. Then the group manager GM chooses at random $\beta_j \in F_q$ and computes $K_j = \varepsilon_{\beta_j}(K_{j-1})$, where $\varepsilon_{\beta_j}(\cdot)$ denotes a secret-keyed permutation over F_q, with secret key β_j. Let $R_j = \{U_{\ell_1}, \ldots, U_{\ell_{w_j}}\}$ be the set of revoked users at session j, satisfying $|R_j| = w_j \leq t$. The group manager broadcasts $B_j = \{R_j, \phi_j(x), \varepsilon_{K_j}(\beta_1), \varepsilon_{K_j}(\beta_2), \ldots, \varepsilon_{K_j}(\beta_j)\}$, where $\phi_j(x) = \Lambda_j(x)K_j + \Psi(x, \alpha_j)$ and $\Lambda_j(x) = (x - \ell_1) \cdot \ldots \cdot (x - \ell_{w_j})$.

Join. When a new user U_k joins the group at some session j, the group manager GM privately sends to him the polynomial $S_k(y) = \Psi(k, y)$ and the value α_j.

Computation of the session key. If a non-revoked user $U_i \notin R_j$ correctly receives the broadcast message B_j for session j, he can compute K_j by first computing α_j (as $\alpha_j = f(\alpha_{j-1})$ if U_i was already in the group at session $j-1$), and then evaluating $S_i(\alpha_j) = \Psi(i, \alpha_j)$, $\phi_j(i)$ and $\Lambda_j(i) \neq 0$. Finally, U_i computes

$$K_j = \frac{\phi_j(i) - \Psi(i, \alpha_j)}{\Lambda_j(i)}.$$

Intuitively, the revoked users $U_{\ell_s} \in R_j$ cannot compute K_j because $\Lambda_j(\ell_s) = 0$, for all $s = 1, \dots, w_j$.

Alternatively, a user U_i who correctly receives B_{j_1} and B_{j_2}, where $1 \leq j_1 < j_2 \leq m$ and U_i is not revoked in session j_2, can still recover the secret session key K_j, for all $j \in \{j_1, \dots, j_2\}$, as follows. From B_{j_2}, the user U_i can recover K_{j_2} (as explained just above) and from this key K_{j_2} and the last values in B_2, he can also recover the values $\beta_1, \beta_2, \dots, \beta_{j_2}$. From B_{j_1}, the user can recover K_{j_1}. After that, the user just computes $K_j = \varepsilon_{\beta_j}(K_{j-1})$, for $j = j_1 + 1, \dots, j_2 - 1$, obtaining in this way all the intermediate keys, between sessions j_1 and j_2.

Dutta and Mukhopadhyay claim that this scheme is an unconditionally secure self-healing key distribution scheme with t-revocation, in particular achieving both forward and backward secrecy (see Theorem 4.2 of [1]).

3 Flaws in the Scheme(s) of Dutta and Mukhopadhyay

In this section we show that the schemes of Dutta and Mukhopadhyay [1, 2, 8], do not satisfy some of the security requirements for self-healing key distribution schemes. For simplicity, we concentrate on the scheme of [1], described in the previous section; but all the attacks that we explain can also be applied to the (similar) schemes in [2], and one of the attacks can be applied to the scheme in [8].

The schemes of Dutta and Mukhopadhyay are *surprisingly* efficient, in the sense that the length of the secret information S_i received by any user is quite smaller than in previous proposals of self-healing key distribution schemes. Actually, this length contradicts the lower bound that Blundo et al. [3, 4] have given for the length of secret information to be stored in a secure self-healing key distribution scheme with t-revocation.

Specifically, Theorem 5.2 of [3] (and similarly, Theorem 4.1 of [4]) states that, for any user U_i belonging to the group since session j, it holds $H(\mathbf{S}_i) \geq (m - j + 1) \log q$, where operator H is the Shannon entropy and $\mathbf{S}_i$ denotes the random variable representing the secret information of user U_i. In other words, a user U_i has to store at least as many bits of secret information as the sum of the bits of all the session keys that U_i might compute as member of the group. However, in the considered scheme of Dutta and Mukhopadhyay [1], each user receives $(t + 2) \log q$ bits of secret information, independently of the number of session keys (up to m) that this user may want to compute. According to the result in [3], a user who wants to recover all the session keys $K_j \in F_q$, $1 \leq j \leq m$, in the scheme of [1] should store a secret information of at least $m \log q$ bits.

Summing up, we have a contradiction between Theorem 5.2 of [3] and the security of the scheme of Dutta and Mukhopadhyay (Theorem 4.2 of [1]). A first justification of this contradiction is easily detectable, because there is a clear mistake in the statement of the security results in [1]. They assert that the security of their schemes is unconditional, meaning that the schemes resist attacks even from computationally unlimited adversaries. However, since the schemes involve the use of a one-way permutation $f : F_q \to F_q$, it is easy to find an attack against the backward secrecy property, executed by an unlimited adversary who can invert the permutation f. This adversary controls a user U_i who joins the group at some session $j > 1$. In the Join phase, he receives α_j along with his secret information S_i. If he has unlimited computational resources, he can invert the one-way permutation f and obtain $\alpha_{j-1} = f^{-1}(\alpha_j)$, $\alpha_{j-2} = f^{-1}(\alpha_{j-1})$ and so on, until he obtains all the values $\alpha_0, \alpha_1, \ldots, \alpha_j$. Combining these values with his secret information S_i and the previous broadcast messages $B_1, \ldots, B_{j-1}$, this user can easily compute all the previous session keys $K_1, \ldots, K_{j-1}$. Therefore, the security (in particular, the backward secrecy) of the schemes can, at most, be computational.

But, in the following sections, we show that the considered schemes are not even computationally secure. We provide explicit attacks against the backward and forward secrecy properties of the scheme of Dutta and Mukhopadhyay.

3.1 A (Fixed) Attack against Backward Secrecy

Actually, the scheme does not achieve even computational backward secrecy, because of the following simple attack. The adversary controls a user U_i who joins the group at some session $j > 1$. Once he has S_i and α_j, he can compute the session key K_j from the broadcast packet B_j. Now, he can obtain the values $\beta_1, \ldots, \beta_j$ by combining K_j with the last elements in the broadcast packet; he must apply the inverse permutation of $\varepsilon_{K_j}(\cdot)$.

From this point on, and due to the fact that $K_j = \varepsilon_{\beta_j}(K_{j-1})$, it is clear that U_i can recover K_{j-1} by inverting ε_{β_j}, and also the other previous keys $K_{j-2}, \ldots, K_1$, by iterating this process. Note that this same attack also applies to another scheme, in [8], proposed by (essentially) the same authors.

This flaw can be fixed, in the following way. The group manager GM chooses values $\beta_1, \ldots, \beta_m$ and $\gamma_1, \ldots, \gamma_m$ at the same time, in the Setup phase. Later, in the Broadcast phase of the j-th session, he defines $K_j = \varepsilon_{\beta_j+\gamma_j}(K_{j-1})$ and defines the last elements of the broadcast B_j to be $\{\varepsilon_{K_j}(\beta_\ell)\}_{\ell=j+1,\ldots,m}$ and $\{\varepsilon_{K_j}(\gamma_\ell)\}_{\ell=1,\ldots,j-1}$. It is quite easy to see that the self-healing scheme resulting from this slight modification resists the above-mentioned attack, and that it satisfies the self-healing and backward secrecy properties, of course in a computational (not in an unconditional) way, because of the aforementioned argument.

3.2 A (Serious) Attack against Forward Secrecy

Finally, we present a more serious attack against the forward secrecy property of the scheme of Dutta and Mukhopadhyay. Let us assume $t + 1 < m$ and let us consider a user $U_i \in G_0$ who remains non-revoked during r sessions, where $t < r < m$. Without loss of generality, we can assume the r sessions are the $t + 1$ first ones.

Then, he knows the keys $K_1, \ldots, K_{t+1}$ and the values $\alpha_1, \ldots, \alpha_{t+1}$ for the first $t + 1$ sessions. In particular, U_i can use the key K_j and the broadcast information $\phi_j(x), R_j$ in B_j to compute $\psi(x, \alpha_j) = \phi_j(x) - \Lambda_j(x)K_j$, for every $j = 1, \ldots, t + 1$.

Therefore, U_i has $t + 1$ tuples of the form $(\alpha_1, \psi(x, \alpha_1)), \ldots, (\alpha_{t+1}, \psi(x, \alpha_{t+1}))$. Thus, since the polynomial ψ has degree t in the second variable, U_i can interpolate (for example, by using Lagrange interpolation) and he gets the complete bivariate polynomial $\psi(x, y)$. Once U_i obtains the polynomial $\psi(x, y)$ he is able to compute keys for future sessions, even if he is revoked for those sessions, contradicting the forward secrecy property of the scheme, asserted by the authors in [1].

For simplicity, let us assume that user U_i is revoked in session $t + 2 \leq m$ and we will show how he can compute K_{t+2}. The key point for that computation is that, since user U_i has been able to compute $\psi(x, y)$, he can calculate $\psi(i^*, y) = S_{i^*}(y)$ for any user U_{i^*} non-revoked in the $(t + 2)$-th session. Furthermore, U_i can also compute $\alpha_{t+2} = f(\alpha_{t+1})$ and so the value $S_{i^*}(\alpha_{t+2}) = \psi(i^*, \alpha_{t+2})$. Then, when the group manager broadcasts in the $(t + 2)$-th session the information $B_{t+2} = \{R_{t+2}, \phi_{t+2}(x), \varepsilon_{K_{t+2}}(\beta_1), \varepsilon_{K_{t+2}}(\beta_2), \ldots, \varepsilon_{K_{t+2}}(\beta_{t+2})\}$, where $\phi_{t+2}(x) = \Lambda_{t+2}(x)K_{t+2} + \Psi(x, \alpha_{t+2})$, user U_i can easily compute the key K_{t+2} for the session $t + 2$ as follows:

$$K_{t+2} = \frac{\phi_{t+2}(i^*) - \Psi(i^*, \alpha_{t+2})}{\Lambda_{t+2}(i^*)}.$$

Note that $\Lambda_{t+2}(i^*) \neq 0$ as we are assuming the user U_{i^*} is not in the list R_{t+2} of revoked users for that session.

An Example

We add here a simple example to illustrate how this attack works. It may be useful to see the variable x as a constant, when thinking on $\psi(x, y)$. We can thus define the polynomial $F(y) = \psi(x, y)$, with $t+1$ unknowns. In the attack explained above, the attacker holds $t+1$ pairs $\{(\alpha_j, F(\alpha_j))\}_{j=1,\ldots,t+1}$, so $F(y) = \psi(x, y)$ can be recovered with polynomial interpolation.

For example, consider $t = 2$, and suppose that the **unknown** polynomial $\psi(x, y)$ is

$$\psi(x, y) = 3-x+2y+xy-2x^2+3y^2-x^2y+2xy^2-3x^2y^2 = (3-x-2x^2)+y(2+x-x^2)+y^2(3+2x-3x^2).$$

Suppose that the attacker can obtain $F(\alpha_j) = \psi(x, \alpha_j)$ for $t + 1 = 3$ different values of α_j, for simplicity $\alpha_j = -1, 0, 1$. This means that the attacker knows

- $F(-1) = \psi(x, -1) = 3 - x - 2x^2 - (2 + x - x^2) + (3 + 2x - 3x^2) = 4 - 4x^2$.
- $F(0) = \psi(x, 0) = 3 - x - 2x^2$.
- $F(1) = \psi(x, 1) = 3 - x - 2x^2 + (2 + x - x^2) + (3 + 2x - 3x^2) = 8 + 2x - 6x^2$.

Now the attacker can use Lagrange interpolation (we define $S = \{-1, 0, 1\}$):

$$\psi(x, y) = F(y) = \sum_{\alpha_j \in S} F(\alpha_j) \prod_{\alpha_k \in S, \alpha_k \neq \alpha_j} \frac{y - \alpha_k}{\alpha_j - \alpha_k}.$$

In our specific case, the attacker can recover $F(y) = \psi(x, y)$ as

$$\psi(x, y) = (4 - 4x^2)\frac{y}{-1}\frac{y-1}{-2} + (3 - x - 2x^2)\frac{y+1}{1}\frac{y-1}{-1} + (8 + 2x - 6x^2)\frac{y+1}{2}\frac{y}{1}.$$

The result of operating this formula is exactly the initial polynomial $\psi(x, y) = 3 - x + 2y + xy - 2x^2 + 3y^2 - x^2y + 2xy^2 - 3x^2y^2$.

4 Conclusion

The self-healing key distribution schemes by Dutta and Mukhopadhyay [1, 2] are the most efficient ones with respect to the length of the secret information held by each user. In this paper we have explained why this 'short' length contradicts some well-known theoretical bounds on the efficiency of self-healing schemes. After that, we have shown some explicit attacks which completely break the security of the considered schemes. Although one of the attacks (which applies also to a different scheme, in [8]) can be fixed, the other one cannot be avoided without decreasing the efficiency of the scheme, due to the theoretical results in [3, 4].

Our personal conclusion is that the efficiency of previous self-healing schemes could not be substantially (and securely) improved, in the case of unconditionally secure schemes, because the schemes proposed up to 2006 already achieve the optimal theoretical bounds for the efficiency parameters.

Acknowledgments

The work of the two first authors is partially supported by Spanish program CONSOLIDER-INGENIO 2010, under projects ARES (CSD2007-00004) and COMONSENS (CSD2008-00010). Javier Herranz enjoys a *Ramón y Cajal* grant, partially funded by the European Social Fund (ESF), from Spanish MICINN Ministry. Finally, the work of Germán Sáez is partially supported by Spanish MICINN Ministry, under project TSI2006-02731.

References

[1] R. Dutta and S. Mukhopadhyay, "Improved self-healing key distribution with revocation in wireless sensor network," *Proc. Wireless Communications and Networking Conference, (WCNC)*, 2007, pp. 2963–2968.

[2] R. Dutta and S. Mukhopadhyay, "Designing scalable self-healing key distribution schemes with revocation capability," *Proc. ISPA*, 2007, Lecture Notes in Computer Science, vol. 4742, pp. 419–430.

[3] C. Blundo, P. D'Arco, A. De Santis and M. Listo, "Design of self-healing key distribution schemes," *Designs, Codes and Cryptography*, vol. 32, pp. 15–44, 2004.

[4] C. Blundo, P. D'Arco and A. De Santis, "On self-healing key distribution schemes," *IEEE Transactions on Information Theory*, vol. 52 (12), pp. 5455–5467, 2006.

[5] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin and D. Dean, "Self-healing key distribution with revocation," *Proc. IEEE S&P*, 2002, pp. 241–257.

[6] D. Liu, P. Ning and K. Sun, "Efficient self-healing key distribution with revocation capability," *Proc. CCS*, 2003, ACM Press, pp. 231–240.

[7] G. Sáez, “On threshold self-healing key distribution schemes,” *Proc. CCC*, 2005, Lecture Notes in Computer Science, vol. 3796, pp. 340–354.

[8] R. Dutta, S. Mukhopadhyay and S. Emmanuel, “Low bandwidth self-healing key distribution for broadcast encryption,” *Proc. 2nd Asia International Conference on Modelling and Simulation (AMS)*, 2008, pp.867-872.