行政院國家科學委員會補助專題研究計畫 □ 成 果 報 告 ☑期中進度報告

具密文與金鑰效率之公開式廣播加密系統之研究

Public-Key Broadcast Encryption with Efficient Ciphertext and Private Keys

計畫類別：☑ 個別型計畫　□整合型計畫
計畫編號：NSC 96-2628-E-009-011-MY3
執行期間： 96 年 8 月 1 日至 99 年 7 月 31 日

計畫主持人：曾文貴 教授
計畫參與人員：林孝盈、林煥宗、簡韶瑩、陳毅睿

成果報告類型(依經費核定清單規定繳交)：☑精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學 資訊工程系

中 華 民 國 98 年 6 月 30 日

中文摘要

本研究計畫研究公開式廣播加密系統，現今最好的公開式廣播加密系統的私密金鑰大小、公開金鑰和傳輸量能無法和私密式的廣播系統相比，我們覺得可以使之達到更佳的效率：分別為 O(r), O(log n) 和 O(1)，同時計算量也可控制在合理的範圍之內，不像 BGW 的方法需要 O(n)。

去年度我們發展出兩個公開廣播加密協定，第一個協定可以達到 O(r)密文長度，O(log n)私密金鑰及 O(1)公開金鑰，計算量需要 O(r)。第二個協定可以達到 O(r)密文長度，$O(\log^2 n)$私密金鑰及 O(1)公開金鑰，計算量只需要 O(1)。論文已在今年的 PKC 會議上發表。

今年度我們將協定修改，加強其安全度達到 IND-CCA2 的等級，目前投稿到知名的期刊，正在審稿中。除此之外，我們還進行了有關感測網路金鑰建立的問題，我們提出一個和現有論文完全不同的攻擊模型，再據此提出一個安全的金鑰建立協定並探討其安全性，目前這篇論文以被知名期刊接受。

關鍵詞：廣播加密、公開金鑰。

英文摘要

In this project we study the public-key broadcast encryption system, in which one can broadcast to a set of authorized users. To our best knowledge, the best public-key broadcast encryption system is not very efficient in the size of the header, public key and private key of users, compared to the secret-key broadcast encryption system. One of the goals of this research is to design and analyze efficient public-key broadcast encryption schemes.

In the last year, we designed two efficient public-key broadcast encryption schemes. The first scheme achieves O(1) public-key size, O(r) header size and O(log n) private keys per user. The decryption time is reasonably O(r). Our second scheme achieves O(1) public-key size, O(r) header size and $O(\log^2 n)$ private keys per user. Although the private key size is less efficient than the first one, its decryption time is remarkably O(1). The paper of these results has been published in prestigious PKC conference.

In this year, we improve one of our designed schemes to achieve the IND-CCA2 security and give a very strict proof. We have submitted the improved result to a prestigious journal. In addition to the work on public-key broadcast encryption, we also work on the key establishment problem in the wireless sensor networks. We explore a different security model in which the adversary is instead storage-bounded, not computing-power constraint. By this model, we propose a very simple and secure key establishment protocol. The protocol does not require the sensors to pre-load secret. This result has been accepted by a prestigious international journal.

Keywords: Broadcast encryption, public key system.

一、計畫緣起及目的

廣播加密是一種有效率的金鑰管理及訊息傳播機制。對於大量的使用者，管理中心可以傳播訊息給任意指定(未被註銷)的使用者，指定的使用者收到訊息後，可依表頭的內容解開資訊；而被註銷的使用者，即使共謀也無法從中得到資訊。廣播加密在生活上有很多應用，如付費電視、線上影片等。廣播加密的正式討論最早是在 1993 年由 Fiat 和 Naor 所提出，在廣播加密的機制中，一開始管理中心會分配每位使用者 u 一些金鑰 k。廣播時，中心首先會使用一把金鑰 SK 對欲傳送的訊息 M 做加密，接著依照接收使用

者的集合，使用某些 k 對金鑰 SK 做加密，此為表頭，連結欲傳送之加密訊息，形成下列的廣播格式：

$$< E_{k_1}(SK), E_{k_2}(SK), \ldots E_{SK}(M) >$$

使用者接收到訊息後，首先利用表頭和所擁有的金鑰來解出 SK，接著用 SK 即可還原訊息 M。

根據使用者一開始所分配的金鑰改變與否，廣播加密可分為有狀態加密機制(stateful)和無狀態加密機制 (stateless)，在無狀態加密機制中，使用者的金鑰分配完成後即不再更改，此法符合許多裝置的限制，大幅的提升了廣播加密的應用性，如使用在 DVD 和 VCD 分區上。無狀態加密機制方法中，又可再區分為私密式廣播加密 及公開式廣播加密系統，其中差別在於私密式廣播加密系統，只有知道所有使用者秘鑰 (如設置中心)才可廣播，而公開式廣播加密則是每人皆可廣播，並只有擁有相對秘鑰者才可解開訊息。

Naor 和 Naor 等人於 2001 年所提出一個可行性高的無狀態私密金鑰廣播加密機制演算法，他們把廣播加密轉換成為 Subset Cover 問題的想法，同時在擬亂數產生器是安全的假設下，利用擬亂數產生器來衍生金鑰，大幅減少使用者金鑰儲存的數量，並突破了原本 Luby 所計算出在完全（unconditionally）安全性上傳輸量和計算量關係的下限。後來許多學者提出了各種架構來改進廣播加密的方法。

公開金鑰廣播加密系統的成果比較少，最早的論文為 Boneth and Franklin 提出，之後 Tzeng and Tzeng 提出用多項式插值的技術來達到剔除使用者與追蹤背叛者 (traitor)的功能，後來 Kurosawa and Yoshida 將其推廣到使用任何 linear error correcting code 皆可。最近 Boneth, Gentry and Waters 提出廣播量和儲存金鑰量都很少的公開金鑰廣播加密的方法，缺點是公開金鑰的量非常大。2003 年，Dodis and Fazio 提出了利用 IBE (identity-based encryption)系統把私密式廣播加密系統轉化成公開式廣播加密的系統的方法，轉換出來的系統的各項參數和原來的私密金鑰系統的皆相同。

在廣播加密之中，重要的參數有下列幾個，第一個是表頭大小 t (Header size)也就是傳輸量，第二則是金鑰的儲存量，第三則是每個使用者所需的計算量。在一些研究中，某些方法會限制註銷使用者共謀的個數，然而在此篇文章中，我們著重在探討無限制註銷者共謀(collusion resistant)的方法。關於金鑰分配和傳輸量之間的關係，直覺的想法，假設現在有 n 位使用者，每位使用者擁有一把自己專屬的金鑰，則當我們註銷掉任意 r 位使用者時，我們需要對其餘 n-r 位使用者一一加密，因此，此方法所需的傳輸量為 n-r，每位使用者金鑰的儲存量則為 1，計算量方面，由於使用者收到後可直接使用金鑰解開表頭，因此計算量也為 1 (以上這種方法我們取名為(a)列於下表之中)。相反的，若我們分給每位使用者 2n-1 把金鑰，每把金鑰分別代表自己之外其餘 n-1 個使用者註銷的情形，則當我們註銷 r 個使用者時，我們所需的傳輸量為 1，每位使用者金鑰儲存量即為 2n-1 (以上這種方法我們取名為(b)列於下表之中)，且我們需要 O(n)的金鑰查詢時間。

由上述兩種方法我們觀察可得知，當每個使用者金鑰儲存量少的時候，傳輸量多；當儲存量少之時，所需傳輸量就大，而如何能有個好方法能在這兩者間取得平衡？亦或是使這兩者參數皆小，並在計算量上所需最小，便是我們研究的主要課題。

二、研究成果

本年度(三年期計畫的第二年)的研究成果為改進第一年提出的一個協定，使其安全性達到最高的 IND-CCA2 安全，目前修訂後的論文已經投稿到知名期刊，正在審查中。詳細的方法請見附件的論文。

除此之外，我們還進行了有關感測網路金鑰建立的問題，我們提出一個和現有論文完全不同的攻擊模型，再據此提出一個安全的金鑰建立協定並探討其安全性，目前這篇論文已被期刊 IEEE Trans. Wireless Communications 接受。這篇論文主要是探討 storage-bounded 攻擊者的模式下，建立節點間金鑰的方法，我們發現節點間不需要事先載入秘密值就可建立安全的通訊金鑰，我們使用了機率式的分析方法來討論金鑰的安全行，我們是第一個在感測網路上使用這個分析方法。

三、計畫成果自評

今年度我們改進了第一年度的成果，達到最高的安全等級，已投稿到知名期刊。我們還發表了有關感測網路的金鑰建立成果在 IEEE Trans. Wireless Communications 期刊。以成果來看，我們達成了本計畫第二年度的目的。

參考文獻

1. N. Attrapadung, H. Imai. Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In *Proceedings of Advances in Cryptology - Asiacrypt 05*, Lecture Notes in Computer Science 3788, pp.100-120, Springer, 2005.
2. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proceedings of Advances in Cryptology - Eruocrypt 05*, Lecture Notes in Computer Science 3494, pp.440-456, Springer, 2005.
3. D. Boneh, M. Franklin. An efficient public key traitor tracing scheme. In *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.338-353, Springer, 1999.
4. D. Boneh, C. Gentry, B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of Advances in Cryptology -Crypto 05*, Lecture Notes in Computer Science 3621, pp.258-275, Springer, 2005.
5. D. Boneh, B. Waters. A fully collusion resistant broadcast trace, and revoke system. In *Proceedings of the ACM Conference on Computer and Communications Security - CCS 06*, pp.211-220, ACM Press, 2006.
6. Y. Dodis, N. Fazio. Public key broadcast encryption for stateless receivers. In *Proceedings of Digital Right Management 02 - DRM 02*, Lecture Notes in Computer Science 2696, pp.61-80, Springer, 2002.
7. Y. Dodis, N. Fazio. Public key broadcast encryption secure against adaptive chosen ciphertext attack. In *Proceedings of Public Key Cryptography - PKC 03*, Lecture Notes in Computer Science 2567, pp.100-115, Springer, 2003.
8. A. Fiat, M. Naor. Broadcast encryption. In *Proceedings of Advances in Cryptology - Crypto 93*, Lecture Notes in Computer Science 773, pp.480-491, Springer, 1993.
9. E. Fujisaki, T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.537-554, Springer, 1999.
10. D. Galindo. Boneh-Franklin identity based encryption revisited. In Proceedings of ICALP 05, Lecture Notes in Computer Science 3580, pp.791-802, Springer, 2005.
11. M.T. Goodrich, J.Z. Sun, R. Tamassia. Efficient Tree-Based Revocation in Groups of Low-State Devices. In Proceedings of Advances in Cryptology - Crypto 04, Lecture Notes in Computer Science 3152, pp.511-527, Springer, 2004.
12. D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Proceedings of Advances in Cryptology - Crypto 02*, Lecture Notes in Computer Science 2442, pp.47-60, Springer, 2002.
13. K. Kurosawa, Y. Desmedt. Optimum traitor tracing and symmetric schemes. In *Proceedings of Advances in Cryptology - Eurocrypt 98*, Lecture Notes in Computer Science 1403, pp.145-157, Springer, 1998.
14. K. Kurosawa, T. Yoshida. Linear code implies public-key traitor tracing. In *Proceedings of Public Key Cryptography*, Lecture Notes in Computer Science 2274,

pp.172-187, Springer, 2002.

15. J.W. Lee, Y.H. Hwang, P.J. Lee. Efficient public key broadcast encryption using identifier of receivers. In *Proceedings of International Conference on Information Security Practice and Experience - ISPEC 06*, Lecture Notes in Computer Science 3903, pp.153-164, Springer, 2006.
16. D. Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of Advances in Cryptology - Crypto 01*, Lecture Notes in Computer Science 2139, pp.41-62, Springer, 2001.
17. M. Naor, B. Pinkas. Efficient trace and revoke schemes. In *Proceedings of Financial Cryptography 00*, Lecture Notes in Computer Science 1962, pp.1-20, Springer, 2000.
18. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), pp.612-613, 1979.
19. W.-G. Tzeng, Z.-J. Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In *Proceedings of Public Key Cryptography - PKC 01*, Lecture Notes in Computer Science 1992, pp.207-224, Springer, 2001.
20. P. Wang, P. Ning, D.S. Reeves. Storage-efficient stateless group key revocation. In *Proceedings of the 7th Information Security Conference - ISC 04*, Lecture Notes in Computer Science 3225, pp.25-38, Springer, 2005.
21. E.S. Yoo, N.-S. Jho, J.J. Cheon, M.-H. Kim. Efficient broadcast encryption using multiple interpolation methods. In *Proceedings of ICISC 04*, Lecture Notes in Computer Science 3506, pp.87-103, Springer, 2005.
22. M. Yoshida, T. Fujiwara. An efficient traitor tracing scheme for broadcast encryption. In *Proceedings of 2000 IEEE International Symposium on Information Theory*, pp.463, IEEE Press, 2000.

Public Key Broadcast Encryption with Low Number of Keys and Constant Decryption Time

Yi-Ru Liu, Wen-Guey Tzeng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan 30050
Email: wgtzeng@cs.nctu.edu.tw

***Abstract*—In this paper we propose two public key BE schemes that have efficient complexity measures. The first scheme, called the PkBE-PI scheme, has $O(r)$ header size, $O(1)$ public keys and $O(\log N)$ private keys per user, where r is the number of revoked users. This is the first public key BE scheme that has both public and private keys under $O(\log N)$ while the header size is $O(r)$. These complexity measures match those of efficient secret key BE schemes.**

Our second scheme, called the PkBE-SD-PI scheme, has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys per user also. Its decryption time is remarkably $O(1)$. This is the first public key BE scheme that has $O(1)$ decryption time while other complexity measures are kept low. Overall, this is the most efficient public key BE scheme up to now.

Our basic schemes are one-way secure against *full collusion of revoked users* in the random oracle model under the BDH assumption. We modify our schemes to have indistinguishability security against adaptive chosen ciphertext attacks under the Gap-BDH assumption.

Keywords: Broadcast encryption, public-key system, polynomial interpolation, collusion.

I. Introduction

Assume that there is a set $\mathcal{U}$ of N users. We would like to broadcast a message to a subset S of them such that only the (authorized) users in S can obtain the message, while the (revoked) users not in S cannot get information about the message. Broadcast encryption is a bandwidth-saving method to achieve this goal via cryptographic key-controlled access. In broadcast encryption, a dealer sets up the system and assigns each user a set of private keys such that the broadcasted messages can be decrypted by authorized users only. Broadcast encryption has many applications, such as pay-TV systems, encrypted file sharing systems, digital right management, content protection of recordable data, etc.

A broadcasted message M is sent in the form $\langle Hdr(S,m), \mathcal{E}_m(M)\rangle$, where m is a session key for encrypting M via a symmetric encryption method $\mathcal{E}$. An authorized user in S can use his private keys to decrypt the session key m from $Hdr(S,m)$. Since the size of $E_m(M)$ is pretty much the same for all broadcast encryption schemes, we are concerned about the header size. The performance measures of a broadcast encryption scheme are the header size, the number of private keys held by each user, the size of public parameters of the system (public keys), the time for encrypting a message, and the time for decrypting the header by an authorized user. A broadcast encryption scheme should be able to resist the collusion attack from revoked users. A scheme is *fully collusion-resistant* if even all revoked users collude, they get no information about the broadcasted message.

Broadcast encryption schemes can be stateless or stateful. For a stateful broadcast encryption scheme, the private keys of a user can be updated from time to time, while the private keys of a user in a stateless broadcast encryption scheme remain the same through the lifetime of the system. Broadcast encryption schemes can also be public key or secret key. For a public key BE scheme, any one (broadcaster) can broadcast a message to an arbitrary group of authorized users by using the public parameters of the system, while for a secret key broadcast encryption scheme, only the special dealer, who knows the system secrets, can broadcast a message.

In this paper we refer "stateless public key broadcast encryption" as "public key BE".

A. Our Contribution

We propose two public key BE schemes that have efficient complexity measures. The first scheme, called the PkBE-PI scheme (broadcast encryption with polynomial interpolation), has $O(r)$ header size, $O(1)$ public keys, and $O(\log N)$ private keys per user[1], where r is the number of revoked users. This is the first public key BE scheme that has both public and private keys under $O(\log N)$ while the header size is $O(r)$. These complexity measures match those of efficient secret key BE schemes [11], [20], [21]. The idea is to run $\log N$ copies of the basic scheme in [17], [19], [22] in parallel for lifting the restriction on a priori fixed number of revoked users. Nevertheless, if we implement the $\log N$ copies straightforwardly, we would get a scheme of $O(N)$ public keys. We are able to use the properties of bilinear maps as well as special private key assignment to eliminate the need of $O(N)$ public keys and make it a constant number.

Our second scheme, called the PkBE-SD-PI scheme (public key SD broadcast encryption with polynomial interpolation), is constructed by combining the polynomial interpolation technique and the subset cover method in the SD scheme [16]. The PkBE-SD-PI scheme has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys per user. They are comparable to those of the PkBE-PI scheme. Nevertheless, the decryption

[1] log is based on 2 if the base is not specified.

time is remarkably $O(1)$. This is the first public key broadcast encryption scheme that has $O(1)$ decryption time while other complexity measures are kept low.

Our basic schemes are one-way secure against *full collusion of revoked users* in the random oracle model under the BDH assumption. We modify our schemes to have indistinguishability security against adaptive chosen ciphertext attacks under the Gap-BDH assumption. The comparison with some other public key BE schemes with full collusion resistance is shown in Table I.

B. Related Work

Fiat and Naor [9] formally proposed the concept of static secret key broadcast encryption. Many researchers followed to propose various broadcast encryption schemes, e.g., see [11], [12], [16], [17], [20].

Kurosawa and Desmedt [13] proposed a pubic-key BE scheme that is based on polynomial interpolation and traces at most k traitors. The similar schemes of Noar and Pinkas [17], Tzeng and Tzeng [19], and Yoshida and Fujiwara [22] allow revocation of up to k users. Kurosawa and Yoshida [14] generalized the polynomial interpolation (in fact, the Reed-Solomon code) to any linear code for constructing public key BE schemes. The schemes in [8], [13], [14], [17], [19], [22] all have $O(k)$ public keys, $O(1)$ private keys, and $O(r)$ header size, $r \leq k$. However, k is a-priori fixed during the system setting and the public key size depends on it. These schemes can withstand the collusion attack of up to k revoked users only. They are not fully collusion-resistant.

Yoo, et al. [21] observed that the restriction of a pre-fixed k can be lifted by running $\log N$ copies of the basic scheme with different degrees (from 2^0 to N) of polynomials. They proposed a scheme of $O(\log N)$ private keys and $O(r)$ header size such that r is not restricted. However, their scheme is secret key and the system has $O(N)$ secret values. In the public key setting, the public key size is $O(N)$.

Recently Boneh, et al. [4] proposed a public key BE scheme that has $O(1)$ header size, $O(1)$ private keys, and $O(N)$ public keys. By trading off the header size and public keys, they gave another scheme with $O(\sqrt{N})$ header size, $O(1)$ private keys and $O(\sqrt{N})$ public keys. Lee, et al. [15] proposed a better trade-off by using receiver identifiers in the scheme. It achieves $O(1)$ public key, $O(\log N)$ private keys, but, $O(r \log N)$ header size. Boneh and Waters [5] proposed a scheme that has the traitor tracing capability. This type of schemes [4], [5], [15] has the disadvantage that the public keys are needed by a user in decrypting the header. Thus, the de-facto private key of a user is the combination of the public key and his private key.

It is possible to transform a secret key BE scheme into a public key one. For example, Dodis and Fazio [7] transformed the SD and LSD schemes [12], [16] into public key SD and LSD schemes, shorted as PkBE-SD-HIBE and PkBE-LSD-HIBE. The transformation employs the technique of hierarchical identity-based encryption to substitute for the hash function. Instantiated with the newest constant-size hierarchical identity-based encryption [2], the PkBE-SD-HIBE scheme has $O(r)$ header size, $O(1)$ public keys and $O(\log^2 N)$ private keys. The PkBE-LSD-HIBE scheme has $O(r/\epsilon)$ header size, $O(1)$ public keys and $O(\log^{1+\epsilon} N)$ private keys, where $0 < \epsilon < 1$ is a constant. The decryption costs of the PkBE-SD-HIBE and PkBE-LSD-HIBE schemes are both $O(\log N)$, which is the time for key derivation incurred by the original relation of private keys. If we apply the HIBE technique to the secret key BE schemes of $O(\log N)$ or $O(1)$ private keys [1], [11], [20], we would get their public key versions with $O(N)$ private keys and $O(N)$ decryption time.

II. Preliminaries

Bilinear map. We use the properties of bilinear maps. Let G and G_1 be two (multiplicative) cyclic groups of prime order q and $\hat{e}$ be a bilinear map from $G \times G$ to G_1. Then, $\hat{e}$ has the following properties.

1) For all $u, v \in G$ and $x, y \in Z_q$, $\hat{e}(u^x, v^y) = \hat{e}(u, v)^{xy}$.
2) Let g be a generator of G, $\hat{e}(g, g) \neq 1$ is a generator of G_1.

BDH hardness assumption. The BDH problem is to compute $\hat{e}(g, g)^{abc}$ from given (g, g^a, g^b, g^c). We say that BDH is (t, ϵ)-hard if for any probabilistic algorithm A with runtime bound t, there is some k_0 such that for any $k \geq k_0$,

$$\Pr[A(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc} : g \xleftarrow{u} G; a, b, c \xleftarrow{u} Z_q] \leq \epsilon.$$

Gap-BDH hardness assumption. The Gap-BDH problem is to compute $\hat{e}(g, g)^{abc}$ from given (g, g^a, g^b, g^c) by accessing to the decision oracle $\mathcal{O}_{\text{BDDH}}$ of indicating whether an input $(g_1, g_2, g_3, g_4, g_5)$ satisfying $\log_{g_1} g_2 \cdot \log_{g_1} g_3 \cdot \log_{g_1} g_4 = \log_{\hat{e}(g_1, g_1)} g_5$. We say that the Gap-BDH problem is $(t, \epsilon, q_{\text{BDDH}})$-hard if for any probabilistic algorithm A with runtime bound t and asking at most q_{BDDH} queries, there is some k_0 such that for any $k \geq k_0$,

$$\Pr[A^{\mathcal{O}_{\text{BDDH}}}(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc} : g \xleftarrow{u} G; a, b, c \xleftarrow{u} Z_q] \leq \epsilon.$$

Broadcast encryption. A public key BE scheme Π consists of three probabilistic polynomial-time algorithms:

- Setup(1^ω, ID, $\mathcal{U}$). Wlog, let $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$. It takes as input the security parameter ω, a system identity ID and a set $\mathcal{U}$ of users and outputs a public key PK and N private key sets $SK_1, SK_2, \ldots, SK_N$, one for each user in $\mathcal{U}$.
- $Enc(PK, S, M)$. It takes as input the public key PK, a set $S \subseteq \mathcal{U}$ of authorized users and a message M and outputs a pair $\langle Hdr(S, m), C \rangle$ of the ciphertext header and body, where m is a randomly generated session key and C is the ciphertext of M encrypted by m via some standard symmetric encryption scheme, e.g., AES.
- $Dec(SK_k, Hdr(S, m), C)$. It takes as input the private key SK_k of user U_k, the header $Hdr(S, m)$ and the body C. If $U_k \in S$, it computes the session key m and then uses m to decrypt C for the message M. If $U_k \notin S$, it cannot decrypt the ciphertext.

TABLE I
COMPARISON OF SOME FULLY COLLUSION-RESISTANT PUBLIC KEY BE SCHEMES.

	header size	public-key size	private-key size	decryption cost$^\natural$
PkBE-SD-HIBE†	$O(r)$	$O(1)$	$O(\log^2 N)$	$O(\log N)$
BGW-I [4]	$O(1)$	$O(N)^\flat$	$O(1)$	$O(N-r)$
BGW-II [4]	$O(\sqrt{N})$	$O(\sqrt{N})^\flat$	$O(1)$	$O(\sqrt{N})$
BW[5]	$O(\sqrt{N})$	$O(\sqrt{N})^\flat$	$O(\sqrt{N})$	$O(\sqrt{N})$
LHL§ [15]	$O(rD)$	$O(2C)^\flat$	$O(D)$	$O(C)$
P-NP, P-TT, P-YF‡	$O(r)$	$O(N)$	$O(\log N)$	$O(r)$
Our work: PkBE-PI	$O(r)$	$O(1)$	$O(\log N)$	$O(r)$
Our work: PkBE-SD-PI	$O(r)$	$O(1)$	$O(\log^2 N)$	$O(1)$

N - the number of users.
r - the number of revoked users.
† - the transformed SD scheme [7] instantiated with constant-size HIBE [2].
‡ - the parallel extension of [17], [19], [22].
$^\flat$ - the public keys are needed for decrypting the header by a user.
§ - $N = C^D$.
$^\natural$ - group operation/modular exponentiation and excluding the time for scanning the header.

The system is correct if all users in S can get the broadcasted message M.

Security. We describe the indistinguishability security against adaptive chosen ciphertext attacks (IND-CCA security) for broadcast encryption as follows [4]. Here, we focus on the security of the session key, which in turn guarantees the security of the ciphertext body C. Let Enc^* and Dec^* be like Enc and Dec except that the message M and the ciphertext body C are omitted. The security is defined by an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ via the following game.

Init. The adversary $\mathcal{A}$ chooses a system identity ID and a target set $S^* \subseteq \mathcal{U}$ of users to attack.
Setup. The challenger $\mathcal{C}$ runs Setup(1^ω, ID, $\mathcal{U}$) to generate a public key PK and private key sets $SK_1, SK_2, \ldots, SK_N$. The challenger $\mathcal{C}$ gives SK_i to $\mathcal{A}$, where $U_i \notin S^*$.
Query phase 1. The adversary $\mathcal{A}$ issues decryption queries Q_i, $1 \le i \le n$, of form $(U_k, S, Hdr(S,m))$, $S \subseteq S^*$, $U_k \in S$, and the challenger $\mathcal{C}$ responds with $Dec^*(SK_k, Hdr(S,m))$, which is the session key encrypted in $Hdr(S,m)$.
Challenge. The challenger $\mathcal{C}$ runs $Enc^*(PK, S^*)$ and outputs $y = Hdr(S^*, m)$, where m is randomly chosen. Then, $\mathcal{C}$ chooses a random bit b and a random session key m^* and sets $m_b = m$ and $m_{1-b} = m^*$. $\mathcal{C}$ gives $(m_0, m_1, Hdr(S^*, m))$ to $\mathcal{A}$.
Query phase 2. The adversary $\mathcal{A}$ issues more decryption queries Q_i, $n+1 \le i \le q_D$, of form (U_k, S, y'), $S \subseteq S^*, U_k \in S, y' \neq y$, and the challenger $\mathcal{C}$ responds with $Dec^*(SK_k, y')$.
Guess. $\mathcal{A}$ outputs a guess b' for b.

In the above the adversary A is static since it chooses the target set S^* of users before the system setup. Let $\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(\omega)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

$$\begin{aligned}&\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(\omega) = \\ &\quad |2 \cdot \Pr[\mathcal{A}^{\mathcal{O}}(PK, SK_{\mathcal{U}\setminus S^*}, m_0, m_1, Hdr(S^*,m)) = b : \\ &\quad S^* \subseteq \mathcal{U}, (PK, SK_{\mathcal{U}}) \leftarrow Setup(1^\omega, \mathrm{ID}, \mathcal{U}), \\ &\quad Hdr(S^*,m) \leftarrow Enc^*(PK, S^*), b \xleftarrow{u} \{0,1\}] - 1|,\end{aligned}$$

where $SK_{\mathcal{U}} = \{SK_i : 1 \le i \le N\}$ and $SK_{\mathcal{U}\setminus S^*} = \{SK_i : U_i \notin S^*\}$.

Definition 1: A public key BE scheme Π=(Setup, Enc, Dec) is (t, ϵ, q_D)-IND-CCA secure if for all t-time bounded adversary $\mathcal{A}$ that makes at most q_D decryption queries, we have $\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(\omega) < \epsilon$.

In this paper we first give schemes with one-way security against chosen plaintext attacks (OW-CPA security) and then modify them to be IND-CCA secure by the technique in [6]. The OW-CPA security is defined as follows.

Init. The adversary $\mathcal{A}$ chooses a system identity ID and a target set $S^* \subseteq \mathcal{U}$ of users to attack.
Setup. The challenger $\mathcal{C}$ runs Setup(1^ω, ID, $\mathcal{U}$) to generate a public key PK and private key sets $SK_1, SK_2, \ldots, SK_N$. The challenger $\mathcal{C}$ gives SK_i to $\mathcal{A}$, where $U_i \notin S^*$.
Challenge. The challenger $\mathcal{C}$ runs $Enc^*(PK, S^*)$ and outputs $Hdr(S^*, m)$, where m is randomly chosen.
Guess. $\mathcal{A}$ outputs a guess m' for m.

Since $\mathcal{A}$ can always encrypt a chosen plaintext by himself, the oracle of encrypting a chosen plaintext does not matter in the definition. Let $\mathrm{Adv}^{ow\text{-}cpa}_{\mathcal{A},\Pi}(\omega)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

$$\begin{aligned}&\mathrm{Adv}^{ow\text{-}cpa}_{\mathcal{A},\Pi}(\omega) = \\ &\quad \Pr[\mathcal{A}(PK, SK_{\mathcal{U}\setminus S^*}, Hdr(S^*,m)) = m : S^* \subseteq \mathcal{U}, \\ &\quad (PK, SK_{\mathcal{U}}) \leftarrow Setup(1^\omega, \mathrm{ID}, \mathcal{U}), \\ &\quad Hdr(S^*,m) \leftarrow Enc^*(PK, S^*)].\end{aligned}$$

Definition 2: A public key BE scheme Π=(Setup, Enc, Dec) is (t, ϵ)-OW-CPA secure if for all t-time bounded adversary $\mathcal{A}$, we have $\mathrm{Adv}^{ow\text{-}cpa}_{\mathcal{A},\Pi}(\omega) < \epsilon$.

III. The PKBE-PI Scheme

Let G and G_1 be the bilinear groups with the pairing function $\hat{e}$, where q is a large prime. Let $H_1, H_2 : \{0,1\}^* \to G_1$ be two hash functions and E be a symmetric encryption with key space G_1.

The idea of our construction is as follows. For a polynomial $f(x)$ of degree t, we assign each user U_i a share $f(i)$. The secret is $f(0)$. We can compute the secret $f(0)$ from any $t+1$ shares. If we want to revoke t users, we broadcast their shares. Any non-revoked user can compute the secret $f(0)$ from his own share and the broadcasted ones, totally $t+1$ shares. On the other hand, any collusion of revoked users cannot compute the secret $f(0)$ since they have t shares only, including the broadcasted ones. If less than t users are revoked, we broadcast the shares of some dummy users such that t shares are broadcasted totally. In order to lift the restriction on the number of revoked users, we use $\log N$ polynomials $f_i(x) = \sum_{j=0}^{2^i} a_{i,j}x^j \pmod q$, each for a range of the number of revoked users.

1) **Setup**(1^ω, ID, $\mathcal{U}$): ω is the security parameter, ID is the identity name of the system, and $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$ is the set of users in the system. Wlog, let N be a power of 2. Then, the system dealer does the following:
   - Choose a generator g of group G, and let $\lg = \log_g$ and $g_1 = \hat{e}(g,g)$.
   - Compute $h_i = H_1(\text{ID}\|i)$ for $1 \le i \le \log N$.
   - Compute $g^{a_{i,j}} = H_2(\text{ID}\|i\|j)$ for $0 \le i \le \log N$ and $0 \le j \le 2^i$.
     Remark. The underlying polynomials are, $0 \le i \le \log N$,
     $$f_i(x) = \sum_{j=0}^{2^i} a_{i,j}x^j \pmod q.$$
     The system dealer does not know the coefficients $a_{i,j} = \lg H_2(\text{ID}\|i\|j)$. But, this does not matter.
   - Randomly choose a secret $\rho \in Z_q$ and compute g^ρ.
   - Publish the public key $PK = (\text{ID}, H_1, H_2, \mathcal{E}, G, G_1, \hat{e}, g, g^\rho)$.
   - Assign a set $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\log N}\}$ of private keys to user U_k, $1 \le k \le N$, where
     $$s_{k,i} = (g^{r_{k,i}}, g^{r_{k,i}f_i(k)}, g^{r_{k,i}f_i(0)}h_i^\rho)$$
     and $r_{k,i}$ is randomly chosen from Z_q, $1 \le i \le \log N$.
2) $\mathbf{Enc}(PK, S, M)$: $S \subseteq \mathcal{U}$, $R = \mathcal{U}\backslash S = \{U_{i_1}, U_{i_2}, \ldots, U_{i_l}\}$ is the set of revoked users, where $l \ge 1$. M is the sent message. The broadcaster does the following:
   - Let $\alpha = \lceil \log l \rceil$ and $L = 2^\alpha$.
   - Compute $h_\alpha = H_1(\text{ID}\|\alpha)$.
   - Randomly select distinct $i_{l+1}, i_{l+2}, \ldots, i_L > N$. These $U_{i_t}, l+1 \le t \le L$, are dummy users.
   - Randomly select a session key $m \in G_1$.
   - Randomly select $r \in Z_q$ and compute, $1 \le t \le L$,
     $$g^{rf_\alpha(i_t)} = (\prod_{j=0}^{L} H_2(\text{ID}\|\alpha\|j)^{i_t^j})^r.$$
   - The ciphertext header $Hdr(S,m)$ is
     $$(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), \ldots, (i_L, g^{rf_\alpha(i_L)})).$$
   - The ciphertext body is $C = \mathcal{E}_m(M)$.
3) $\mathbf{Dec}(SK_k, Hdr(S,m), C)$: $U_k \in S$. The user U_k does the following.
   - Compute $b_0 = \hat{e}(g^r, g^{r_{k,\alpha}f_\alpha(k)}) = g_1^{rr_{k,\alpha}f_\alpha(k)}$.
   - Compute $b_j = \hat{e}(g^{r_{k,\alpha}}, g^{rf_\alpha(i_j)}) = g_1^{rr_{k,\alpha}f_\alpha(i_j)}$, $1 \le j \le L$.
   - Use the Lagrange interpolation method to compute
     $$g_1^{rr_{k,\alpha}f_\alpha(0)} = \prod_{j=0}^{L} b_j^{\lambda_j}, \tag{1}$$
     where $\lambda_j = \frac{(-i_0)(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j-i_0)(i_j-i_1)\cdots(i_j-i_{j-1})(i_j-i_{j+1})\cdots(i_j-i_L)} \pmod q$, $i_0 = k$.
   - Compute the session key
     $$\frac{m\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{rr_{k,\alpha}f_\alpha(0)}}{\hat{e}(g^r, g^{r_{k,\alpha}f_\alpha(0)}h_\alpha^\rho)} = \frac{m\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{rr_{k,\alpha}f_\alpha(0)}}{\hat{e}(g^r, h_\alpha^\rho) \cdot g_1^{rr_{k,\alpha}f_\alpha(0)}} = m. \tag{2}$$
   - Use m to decrypt the ciphertext body C to obtain the message M.

Correctness. We can easily see that the scheme is correct by Equation (2).

A. Performance Analysis

For each system, the public key is $(\text{ID}, H_1, H_2, \mathcal{E}, G, G_1, \hat{e}, g, g^\rho)$, which is of size $O(1)$. Since all systems can use the same $(H, \mathcal{E}, G, G_1, \hat{e}, g)$, the public key specific to a system is simply (ID, g^ρ). Each system dealer has a secret ρ for assigning private keys to its users. Each user U_k holds private keys $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\log N}\}$, each corresponding to a share of polynomial f_i in the masked form, $0 \le i \le \log N$. The number of private keys is $O(\log N)$. When r users are revoked, we choose the polynomial f_α of degree 2^α for encrypting the session key, where $2^{\alpha-1} < r \le 2^\alpha$. Thus, the header size is $O(2^\alpha) = O(r)$. It is actually no more than $2r$.

To prepare a header, the broadcaster needs to compute one pairing function, $2^\alpha + 2$ hash functions, and $2^\alpha + 2$ modular exponentiations, which is $O(r)$ modular exponentiations.

For a user in S to decrypt a header, with a little rearrangement of Equation (1) as
$$\prod_{j=0}^{L} b_j^{\lambda_j} = b_0^{\lambda_0} \cdot \hat{e}(g^{r_{k,\alpha}}, \prod_{j=1}^{L} (g^{rf_\alpha(i_j)})^{\lambda_j}),$$
the user needs to perform 3 pairing functions and 2^α modular exponentiations, which is $O(r)$ modular exponentiations. The evaluation of λ_j's can be done in $O(L) = O(2r)$ if the header consists of
$$\tilde{\lambda}_j = \frac{(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j-i_1)\cdots(i_j-i_{j-1})(i_j-i_{j+1})\cdots(i_j-i_L)} \bmod q,$$

$1 \leq j \leq L$. The user can easily compute λ_j's from $\tilde{\lambda}_j$'s. Inclusion of $\tilde{\lambda}_j$'s in the header does not affect the order of the header size.

B. Security Analysis

We show that it has OW-CPA security in the random oracle model under the BDH assumption.

Theorem 1: Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PkBE-PI scheme is $(t_1 - t', \epsilon_1)$-OW-CPA secure in the random oracle model, where t' is some polynomially bounded time.

Proof: We reduce the BDH problem to the problem of computing the session key from the header by the revoked users. Since the polynomials $f_i(x) = \sum_{j=0}^{L} a_{i,j}x^j$ and secret shares of users for the polynomials are independent for different i's, we simply discuss security for a particular α. Wlog, let $R = \{U_1, U_2, \ldots, U_L\}$ be the set of revoked users and the target set of attack be $S^* = \mathcal{U} \backslash R$. Note that S^* was chosen by the adversary in the **Init** stage. Let the input of the BDH problem be (g, g^a, g^b, g^c), where the pairing function is implicitly known. We set the system parameters as follows:

1) Randomly select $\tau, \mu_1, \mu_2, \ldots, \mu_L, w_1, w_2, \ldots, w_L \in Z_q$.
2) Set the public key of the system:
   a) Let the input g be the generator g in the system.
   b) Set $g^\rho = g^a$.
   c) The public key is $(\text{ID}, H_1, H_2, \mathcal{E}, G, G_1, \hat{e}, g, g^a)$.
   d) The following is implicitly computed.
      - Set $f_\alpha(j) = w_j, 1 \leq j \leq L$.
      - Let $g^{a_{\alpha,0}} = g^{f_\alpha(0)} = g^a \cdot g^\tau = g^{a+\tau}$.
      - Compute $g^{a_{\alpha,i}}$, $1 \leq i \leq L$, from $g^{a_{\alpha,0}}$ and $g^{f_\alpha(j)} = g^{w_j}, 1 \leq j \leq L$, by the Lagrange interpolation method over exponents.
      - Set $h_\alpha = g^b$.
      - For $j \neq \alpha$, choose a random polynomial $f_j(x)$ and set $h_j = g^{z_j}$, where z_j is randomly chosen from Z_q.
3) Set the secret keys $(g^{r_{i,j}}, g^{r_{i,j}f_j(i)}, g^{r_{i,j}f_j(0)}h_j^\rho)$, $0 \leq j \leq \log N$, of the revoked user $U_i, 1 \leq i \leq L$, as follows:
   a) For $j = \alpha$, let $g^{r_{i,\alpha}} = g^{-b+\mu_i}$, $g^{r_{i,\alpha}f_\alpha(i)} = (g^{r_{i,\alpha}})^{w_i}$, and $g^{r_{i,\alpha}f_\alpha(0)}h_\alpha^\rho = g^{(-b+\mu_i)(a+\tau)}(g^b)^a = g^{a\mu_i - b\tau + \mu_i\tau}$.
   b) For $j \neq \alpha$, randomly choose $r_{i,j} \in Z_q$ and compute $g^{r_{i,j}}$, $g^{r_{i,j}f_j(i)}$ and $g^{r_{i,j}f_j(0)}h_j^\rho = g^{r_{i,j}f_j(0)}(g^a)^{z_j}$.
4) Set the header $(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (1, g^{rf_\alpha(1)}), (2, g^{rf_\alpha(2)}), \ldots, (L, g^{rf_\alpha(L)}))$ as follows:
   a) Let $g^r = g^c$.
   b) Compute $g^{rf_\alpha(i)} = (g^c)^{w_i}, 1 \leq i \leq L$.
   c) Randomly select $y \in G_1$ and set $m\hat{e}(g^\rho, h_\alpha)^r = y$. We do not know what m is. But, this does not matter.

Assume that the revoked users together can compute the session key m. During computation, the users can query H_1 and H_2 hash oracles. If the query is of the form $H_2(\text{ID}\|i\|j)$ or $H_1(\text{ID}\|i)$, we set them to be $g^{a_{i,j}}$ and h_i, respectively. If the query has ever been asked, we return the stored hash value for the query. For other non-queried inputs, we return random hash values in G and record them and their hash values.

We should check whether the distributions of the parameters in our reduction and those in the system are equal. We only check those related to α since the others are correctly distributed. Since $\tau, w_1, w_2, \ldots, w_L$ are randomly chosen, $g^{a_{\alpha,i}}, 0 \leq i \leq L$ are uniformly distributed over G^{L+1}. Due to the random oracle model, their corresponding system parameters are also uniformly distributed over G^{L+1}. Since $\mu_1, \mu_2, \ldots, \mu_L$ are randomly chosen, the distribution of h_α and $g^{r_{i,\alpha}}, 1 \leq i \leq L$, are uniform over G^{L+1}, which is again the same as that of the corresponding system parameters. The distributions of g^r in the header and g^ρ in the public key are both uniform over G since they are set from the given input g^c and g^a, respectively. Since the session key m is chosen randomly from G_1, $m\hat{e}(g^\rho, h_\alpha)^r$ is distributed uniformly over G_1. We set it to a random value $y \in G_1$. Even though we don't know about m, it does not affect the reduction. Other parameters are dependent on what have been discussed. We can check that they are all computed correctly. So, the reduction preserves the right distribution.

If the revoked users compute m from the header with probability ϵ, we can solve the BDH problem with the same probability $\epsilon = \epsilon_1$ by computing $y \cdot m^{-1} = \hat{e}(g, g)^{abc}$.

Let t' be the time for this reduction and the solution computation. We can see that t' is polynomially bounded. Thus, if the collusion attack of the revoked users takes $t_1 - t'$ time, we can solve the BDH problem within time t_1. ■

IV. The PkBE-PI Scheme with IND-CCA Security

In Theorem 1, we show that the session key in the header is one-way secure against any collusion of revoked users. In this section, we present an IND-CCA secure PkBE-PI Π' scheme based on the technique in [6]. The scheme has tight security reduction in the success probability.

Let $\Phi : \mathcal{N} \times G \times G_1 \to G_1$ and $\Psi : \mathcal{N} \times G_1 \to G_1$ be two hash functions, modeled as random oracles. Let $\pi : Z_q \to Z_q$ be a collision-resistant hash function. The modification of Π for Π' is as follows.

- In the **Setup** algorithm, add Φ, Ψ, π to PK.
- In the **Enc** algorithm:
  - Compute $A = \hat{e}(g^\rho, h_\alpha)^r$, where $r \in_R Z_q$.
  - Compute $B = m \oplus \Psi(\alpha, A)$.
  - Compute $C = \hat{e}(g^\rho, h_\alpha)^{r/\pi(B)}$.
  - Compute $D_0 = g^{r/\pi(B)}, D_1 = (i_1, g^{rf_\alpha(i_1)/\pi(B)}), D_2 = (i_2, g^{rf_\alpha(i_2)/\pi(B)}), \ldots, D_L = (i_L, g^{rf_\alpha(i_L)/\pi(B)})$.
  - Compute $E = \Phi(\alpha, D_0, C) \oplus B$.
  - The ciphertext header is $Hdr(S, m) = (\alpha, E, D_0, D_1, \ldots, D_L)$.
- In the **Dec** algorithm:
  - Use $D_0, D_1, \ldots, D_L$ and the user's share $(g^{r_{k,\alpha}}, g^{r_{k,\alpha}f_\alpha(k)}, g^{r_{k,\alpha}f_\alpha(0)}h_\alpha^\rho)$ to

compute $g_1^{rr_{k,\alpha} f_\alpha(0)/\pi(B)}$ and then compute $C = \hat{e}(g^\rho, h_\alpha)^{r/\pi(B)}$.
- Compute $B = E \oplus \Phi(\alpha, D_0, C)$.
- Compute $A = C^{\pi(B)}$.
- Compute $m = B \oplus \Psi(\alpha, A)$.

Our scheme Π' is IND-CCA-secure in the random oracle model under the Gap-BDH hardness assumption.

Theorem 2: Assume that the Gap-BDH problem is $(t_1, \epsilon_1, q_{\text{BDDH}})$-hard. The scheme Π' is $(t, \epsilon, q_{H_1}, q_{H_2}, q_\Psi, q_\Phi, q_D, q_{BDDH})$-IND-CCA secure in the random oracle model, where q_{H_1}, q_{H_2}, q_Ψ and q_Φ are the numbers of the adversary's queries to the random oracles H_1, H_2, Ψ and Φ, q_D is the number of adversary's decryption queries, and

$$t = t_1 - O(q_{H_1} + q_{H_2} + q_\Psi + q_\Phi + q_D + q_{\text{BDDH}}) \text{ and}$$
$$\epsilon = \epsilon_1 + O(q_{\text{D}}^2/q).$$

Proof: Assume that adversary $\mathcal{A}$ breaks Π' within time t with success probability ϵ and asking at most q_{H_1}, q_{H_2}, q_Ψ, q_Φ hash queries and q_D decryption queries. We construct an algorithm $\mathcal{B}$ for solving the Gap-BDH problem by simulating the attacking environment of $\mathcal{A}$. Let (g, g^a, g^b, g^c) be an instance of the Gap-BDH problem with bilinear groups G and G_1. Wlog, let $R = \{U_1, U_2, \ldots, U_L\}$ be the set of revoked users in the **Init** stage, where $L = 2^\alpha$. $\mathcal{B}$ maintains H_1-, H_2-, Ψ- and Φ-lists for the random orale queries and four extra watch-lists for H_1, H_2, Ψ and Φ. Wlog, we assume that no queries are asked twice by $\mathcal{A}$. Then, $\mathcal{B}$ works as follows:

Setup. $\mathcal{B}$ prepares the public key of the system and the private keys of the revoked users to $\mathcal{A}$.

1) Randomly select $\tau, \mu_1, \mu_2, \ldots, \mu_L, w_1, w_2, \ldots, w_L \in Z_q$.
2) Set $g^\rho = g^a$.
3) The public key is $(\text{ID}, H_1, H_2, \mathcal{E}, G, G_1, \hat{e}, g, g^a)$, where $\text{ID}, H_1, H_2, \mathcal{E}$ are chosen by Π'.
4) Set $f_\alpha(j) = w_j, 1 \le j \le L$.
5) Let $g^{a_{\alpha,0}} = g^{f_\alpha(0)} = g^a \cdot g^\tau = g^{a+\tau}$. Compute $g^{a_{\alpha,i}}$ from $g^{a_{\alpha,0}}$ and $g^{f_\alpha(j)} = g^{w_j}, 1 \le i, j \le L$, by the Lagrange interpolation method over exponents.
6) Put $(\text{ID}, \alpha, k, g^{a_{\alpha,k}})$ to the H_2-watch-list for $0 \le k \le L$.
7) Set $h_\alpha = g^b$ and put $(\text{ID}, \alpha, h_\alpha)$ to the H_1-watch-list.
8) For $j \ne \alpha$, choose a random polynomial $f_j(x)$ and set $h_j = g^{z_j}$, where z_j is randomly chosen from Z_q. Also, put (ID, j, h_j) to the H_1-watch-list and $(\text{ID}, j, k, g^{a_{j,k}})$ to the H_2-watch-list for $0 \le k \le 2^j$.
9) Set the secret keys $(g^{r_{i,j}}, g^{r_{i,j} f_j(i)}, g^{r_{i,j} f_j(0)} h_j^\rho)$, $0 \le j \le \log N$, of the revoked user $U_i, 1 \le i \le L$, as follows:
   a) For $j = \alpha$, let $g^{r_{i,\alpha}} = g^{-b+\mu_i}$, $g^{r_{i,\alpha} f_\alpha(i)} = (g^{r_{i,\alpha}})^{w_i}$, and $g^{r_{i,\alpha} f_\alpha(0)} h_\alpha^\rho = g^{(-b+\mu_i)(a+\tau)} (g^b)^a = g^{a\mu_i - b\tau + \mu_i \tau}$.
   b) For $j \ne \alpha$, randomly choose $r_{i,j} \in Z_q$ and compute $g^{r_{i,j}}$, $g^{r_{i,j} f_j(i)}$ and $g^{r_{i,j} f_j(0)} h_j^\rho = g^{r_{i,j} f_j(0)} (g^a)^{z_j}$.

$\mathbf{H_1}$ and $\mathbf{H_2}$ queries. When $\mathcal{A}$ makes a query to H_1 (or H_2), $\mathcal{B}$ looks for it in the H_1-watch-list (or H_2-watch-list). If it is found, $\mathcal{B}$ returns the stored hash value and removes the entry from the watch-list. Otherwise, $\mathcal{B}$ returns a random hash value and records it in the H_1-list (or H_2-list).

Decryption queries. $\mathcal{A}$ makes the kth decryption query $(\alpha_k, E_k, D_{0,k}, D_{1,k}, \ldots, D_{L_k,k})$, where $L_k = 2^{\alpha_k}$. If any revoked user in R is not revoked in this decryption query, $\mathcal{B}$ uses the secret key of the revoked user to decrypt. Otherwise, $\mathcal{B}$ looks for $(\alpha_j, D_{0,j}, C_j)$ in the Φ-list such that $\alpha_j = \alpha_k$ and $D_{0,j} = D_{0,k}$.

- If $(\alpha_j, D_{0,j}, C_j)$ is not found, return a random m_k as the plaintext and put $(\alpha_k, D_{0,k}, E_k, m_k)$ to the Φ-watch-list.
- If $(\alpha_j, D_{0,j}, C_j)$ is found, check whether $C_j = \hat{e}(g,g)^{\rho \cdot \log_g h_{\alpha_k} \cdot \log_g D_{0,k}}$. This can be done by querying $(g, g^\rho, h_{\alpha_k}, D_{0,k}, C_j)$ to the $\mathcal{O}_{\text{BDDH}}$ oracle.
  - If $\mathcal{O}_{\text{BDDH}}(g, g^\rho, h_{\alpha_k}, D_{0,k}, C_j) = 1$, retrieve $\phi_j = \Phi(\alpha_j, D_{0,j}, C_j)$ from the Φ-list, compute $B_k = E_k \oplus \phi_j$, $A_k = (C_j)^{\pi(B_k)}$ and return $m_k = B_k \oplus \Psi(\alpha_k, A_k)$ as the plaintext.
  - If $\mathcal{O}_{\text{BDDH}}(g, g^\rho, h_{\alpha_k}, D_{0,k}, C_j) = 0$, return a random m_k as the plaintext and put $(\alpha_k, D_{0,k}, E_k, m_k)$ to the Φ-watch-list.

Hash-Φ queries. When $\mathcal{A}$ queries $(\alpha_k, D_{0,k}, C_k)$ to Φ, $\mathcal{B}$ returns a random ϕ_k. In addition, $\mathcal{B}$ tests whether $\mathcal{O}_{\text{BDDH}}(g, g^\rho, h_{\alpha_k}, D_{0,k}, C_k) = 1$. If so, $\mathcal{B}$ looks for $(\alpha_j, D_{0,j}, E_j, m_j)$ with $\alpha_k = \alpha_j$ and $D_{0,k} = D_{0,j}$ in the Φ-watch-list. For every such entry, delete it from the Φ-watch-list, compute $B_j = E_j \oplus \phi_k$, $A_j = C_k^{\pi(B_j)}$ and $\psi_j = m_j \oplus B_j$, and put (α_j, A_j, ψ_j) to the Ψ-watch-list.

Let us see in advance how this query is handled incorrectly. Firstly, since all such E_j's are different, all $A_j = C_k^{\pi(B_j)}$'s are different unless a collision $\pi(B_{j_1}) = \pi(B_{j_2})$ for π is found. This occurs with a negligible probability. Secondly, $\mathcal{B}$ defines $\Psi(\alpha_j, A_j) = m_j \oplus B_j$ in conflict. The case is when $\alpha_{j_1} = \alpha_{j_2}$ and $C_{k_1} \ne C_{k_2}$, but

$$A_{j_1} = C_{k_1}^{\pi(E_{j_1} \oplus \phi_{k_1})} = C_{k_2}^{\pi(E_{j_2} \oplus \phi_{k_2})} = A_{j_2}.$$

We see that ϕ_k's are jointly independent of the C_k's and E_j's and each troubling C_k comes with an entry in the Φ-watch-list. Furthermore, each such entry is from decryption query. Thus, the conflict probability is at most q_D^2/q.

Hash-Ψ queries. When $\mathcal{A}$ queries (α_k, A_k), $\mathcal{B}$ looks for it in the Ψ-watch-list. If it is found, $\mathcal{B}$ returns the stored hash value ψ_k and delete the entry from the watch-list. Otherwise, $\mathcal{B}$ returns a random hash value ψ_k.

Challenge. $\mathcal{A}$ sends two messages M_0 and M_1 to $\mathcal{B}$. $\mathcal{B}$ discards the messages and sends the challenge

$$(\alpha, E^*, D_0^* = g^c, (1, g^{c f_\alpha(1)}), \ldots, (L, g^{c f_\alpha(L)}))$$

to $\mathcal{A}$, where g^c is from the input instance of the Gap-DDH problem and E^* is randomly chosen.

Additional queries. $\mathcal{A}$ asks more queries and $\mathcal{B}$ responds as described before. When B processes queries, he is on the alert of the query $\Phi(\alpha, D_0^*, C^*)$ with $\mathcal{O}_{\text{BDDH}}(g, g^\rho, h_\alpha, D_0^*, C^*) = 1$. As soon as such a query is asked, $\mathcal{B}$ returns C^* as the solution to the input instance of the Gap-DDH problem and

stops. If $\mathcal{A}$ never asks such a query, $\mathcal{B}$ returns a random value for the solution.

Since $g^\rho = g^a, h_\alpha = g^b$ and $D^* = g^c$, if the query $\Phi(\alpha, D_0^*, C^*)$ with $\mathcal{O}_{\text{BDDH}}(g, g^\rho, h_\alpha, D_0^*, C^*) = 1$ is asked, $C^* = g^{abc}$ is the answer. If no such query is asked, $\mathcal{B}$ can succeed with a negligible probability due to the random oracle model. Thus, $\mathcal{B}$'s success probability of solving the Gap-BDH problem is $\epsilon_1 = \epsilon - \delta$, where $\delta = O(q_D^2/q)$ is the probability that $\mathcal{B}$'s simulation fails or a collision for π is found. This case occurs when $\mathcal{B}$ handles the queries to Φ (hence Ψ) inconsistently.

The runtime of $\mathcal{B}$ is the runtime of $\mathcal{A}$ plus the time of handling the queries of $\mathcal{A}$. For each such query, the handling time is dominated by looking for entries from the random oracle lists and watch-lists. We can use the indexing technique such that the handling time for each such query is $O(1)$. Thus, $\mathcal{B}$'s runtime is $t_1 = t + O(q_{H_1} + q_{H_2} + q_\Phi + q_\Psi + q_D + q_{BDDH})$. ■

V. A Public Key SD Scheme

In the paradigm of subset cover for broadcast encryption [16], the system chooses a collection $\mathcal{C}$ of subsets of users such that each set S of users can be covered by some subsets in $\mathcal{C}$, that is, $S = \cup_{i=1}^w S_i$, where $S_i \in \mathcal{C}$ are disjoint, $1 \le i \le w$. Each subset S_j in $\mathcal{C}$ is associated with a subset key k_j. A user is assigned the subset keys of the subsets to which he belongs. To broadcast the session key m to the users in S, we simply encrypt the session key m with the subset keys $k_i, 1 \le i \le w$. Thus, each user in S can decrypt to obtain the session key.

If the subset keys are all independent, each user would hold too many keys. It is preferable that the subset keys have some relations, for example, one can be derived from another. A subset-cover based broadcast encryption scheme plays the art of choosing $\mathcal{C}$ and designing the subset keys for the subsets in $\mathcal{C}$.

For the SD broadcast encryption method, the collection $\mathcal{C}$ of subsets is chosen as follows. Consider a complete binary tree T of $\log N + 1$ levels. Each node is given a distinct number. Each user in $\mathcal{U}$ is associated with a different leaf node in T. We refer to a complete subtree rooted at node i as "subtree T_i". The subset $S_{i,j}$ in $\mathcal{C}$ is the set of users that are in the subtree rooted at node i, but not in the subtree rooted at node j, where node i is an ancestor of node j. The subset key of the subset $S_{i,i}$ is chosen independently and the subset key of $S_{i,j}, i \ne j$, is derived from the subset key of $S_{i,i}$. Thus, for each subtree T_i in which the user is, if the user holds the subset key of $S_{i,j}$, he can derive the subset key of $S_{i,j'}$, where node j' is a descendent of node j. By key derivation, each user holds only $O(\log^2 N)$ keys since each user belongs to $\log N$ subtrees and for each such subtree the user holds $O(\log N)$ keys. However, the decryption time is $O(\log N)$ due to key derivation.

A. Basic PkBE-SD-PI Scheme

We now present our PkBE-SD-PI scheme, which is constructed by using the polynomial interpolation for key derivation on the subset keys. Instead of directly holding the subset key of $S_{i,j}$, the user derives the subset key by using the his private key and the broadcasted information.

The system setup is similar to that of the PkBE-PI scheme. For each subtree T_i of η levels (level 1 to level η from top to bottom), we define the degree-1 polynomials, $2 \le j \le \eta$,

$$f_{i,j}(x) = a_{i,j,1}x + a_{i,j,0} \pmod q,$$

where $a_{i,j,0} = \lg H_2(\text{ID}\|i\|j\|0)$ and $a_{i,j,1} = \lg H_2(\text{ID}\|i\|j\|1)$. Each user U_k in T_i is given the private keys

$$s_{k,i,j} = (g^{r_{k,i,j}}, g^{r_{k,i,j} f_{i,j}(i_j)}, g^{r_{k,i,j} f_{i,j}(0)} h_{i,j}^\rho)$$

for $2 \le j \le \eta$, where $h_{i,j} = H_1(\text{ID}\|i\|j)$ and nodes $i_1, i_2, \ldots, i_\eta$ are the nodes in the path from node i to the leaf node for U_k (including both ends). We can read $s_{k,i,j}$ as the private key of U_k for the jth level of subtree T_i. This key is used to derive the subset key of $S_{i,j'}$ where node j' is in the same level of node j. In Figure 1, the private keys (in the unmasked form) of U_1 and U_3 for subtree T_i with $\eta = 4$ are given.

Recall that in the SD scheme, the collection $\mathcal{C}$ of subsets is

$$\{S_{i,t} : \text{ node } i \text{ is a parent of node } t, i \ne t\},$$

where $S_{i,t}$ denotes the set of users in subtree T_i, but not in subtree T_t. By our design, if the header contains a masked share for $f_{i,j}(t)$, only the user U_k in $S_{i,t}$ can decrypt the header by using his private key $s_{k,i,j}$, that is, the masked form of $f_{i,j}(s)$, for some $s \ne t$. In Figure 1, the share $f_{i,3}(t)$ is broadcasted so that only the users in $S_{i,t}$ can decrypt the header.

For a set R of revoked users, let S_{i_1,t_1}, $S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a found subset cover for $\mathcal{U}\backslash R$, the header is

$$\begin{aligned}(g^r, &(i_1, t_1, g^{r f_{i_1,j_1}(t_1)}, m\hat{e}(g^\rho, h_{i_1,j_1})^r),\\ &\ldots, (i_z, t_z, g^{r f_{i_z,j_z}(t_z)}, m\hat{e}(g^\rho, h_{i_z,j_z})^r))),\end{aligned}$$

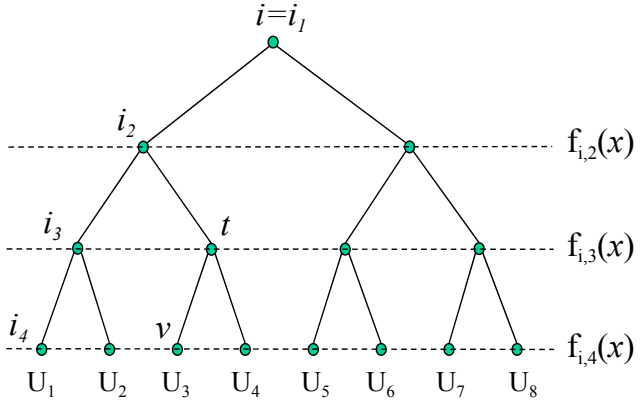
where node t_k is in the j_k-th level of subtree T_{i_k}, $1 \le k \le z$.

For decryption, a non-revoked user finds $(i_k, t_k, g^{r f_{i_k,j_k}(t_k)}, m\hat{e}(g^\rho, h_{i_k,j_k})^r)$ (corresponding to S_{i_k,t_k} where he is in) from the header and applies the Lagrange interpolation to compute the session key m.

Performance. The public key is $O(1)$, which is the same as that of the PkBE-PI scheme. Each user belongs to at most $\log N + 1$ subtrees and each subtree has at most $\log N + 1$ levels. For the subtree of η levels, the user in the subtree holds $\eta - 1$ private keys. Thus, the total number of shares (private keys) held by each user is $\sum_{i=1}^{\log N} i = O(\log^2 N)$. According to [16], the number z of subsets in a subset cover is at most $2|R| - 1$, which is $O(r)$

When the header streams in, a non-revoked user U_k looks for his containing subset S_{i_j,t_j} to which he belongs. With a proper numbering of the nodes in T, this can be done very fast, for example, in $O(\log\log N)$ time. Without considering the time of scanning the header to find out his containing subset, each user needs to perform 2 modular exponentiations and 3 pairing functions. Thus, the decryption cost is $O(1)$.

Security. We first show that the scheme is one-way secure.



– U_1 holds masked shares of $f_{i,2}(i_2)$, $f_{i,3}(i_3)$, $f_{i,4}(i_4)$
– U_3 holds masked shares of $f_{i,2}(i_2)$, $f_{i,3}(t)$, $f_{i,4}(v)$
– For subset $S_{i,t}$, a masked share of $f_{i,3}(t)$ is broadcasted so that U_1, U_2, U_5, U_6, U_7 and U_8 can decrypt, but U_3 and U_4 cannot.

Fig. 1. Level polynomials, private keys and broadcasted shares for subtree T_i.

Theorem 3: Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PkBE-SD-PI scheme is $(t_1 - t', \epsilon_1)$-OW-CPA secure in the random oracle model, where t' is some polynomially bounded time.

Proof: The one-way security proof for the PkBE-SD-PI scheme is similar to that for the PkBE-PI scheme. In the PkBE-SD-PI scheme, all polynomials $f_{i,j}(x)$ are of degree one. Let (g, g^a, g^b, g^c) be the input to the BDH problem. Let $S_{i_1,t_1}, S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a subset cover for $S^* = \mathcal{U} \backslash R$. Recall that the public key of the PkBE-SD-PI method is $(\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^\rho)$. For reduction, we first set $g^\rho = g^a$.

Due to the random oracle assumption for H_1 and H_2, all polynomials are independent. Thus, we can simply consider a particular $S_{\alpha,t}$ in the subset cover for $S^* = \mathcal{U} \backslash R$, where t is at level β of subtree T_α. The corresponding polynomial is $f(x) = f_{\alpha,\beta}(x) = a_1 x + a_0 \pmod q$. Wlog, let $\{U_1, U_2, \ldots, U_l\}$ be the set of revoked users that have the masked share of $f(t)$. The reduction for setting shares of these users $U_i, 1 \le i \le l$, is as follows:

1) Set $f(t) = w$ and compute $g^{f(t)} = g^w$, where w is randomly chosen from Z_q.
2) Let $g^{a_0} = g^{f(0)} = g^a \cdot g^\tau$, where τ is randomly chosen from Z_q.
3) Compute g^{a_1} from $g^{f(t)}$ and g^{a_0} via the Lagrange interpolation.
4) The (random) hash values $H_2(\text{ID} \| \alpha \| \beta \| 0)$ and $H_2(\text{ID} \| \alpha \| \beta \| 1)$ are set as g^{a_0} and g^{a_1} respectively.
5) Set $h_{\alpha,\beta} = g^b \cdot g^{\kappa_{\alpha,\beta}}$, where $\kappa_{\alpha,\beta}$ is randomly chosen from Z_q.
6) The $f(x)$-related secret share of U_i is computed as $(g^{r_i}, g^{r_i f(t)}, g^{r_i f(0)} h_{\alpha,\beta}^\rho)$, where $g^{r_i} = g^{-b} \cdot g^{\mu_i}$ and μ_i is randomly chosen from Z_q. Note that $g^{r_i f(0)} h_{\alpha,\beta}^\rho = g^{a(\mu_i + \kappa_{\alpha,\beta}) - b\tau + \mu_i \tau}$ can be computed from the setting in the previous steps.
7) For non-$f(x)$-related secret shares of U_i (the polynomial is $f_{\alpha,\beta'}(x)$), we set $h_{\alpha,\beta'} = g^{z_{\alpha,\beta'}}$ and compute the shares accordingly, where $z_{\alpha,\beta'}$ is randomly chosen from Z_q.

Finally, we set the challenge as

$$(g^c, (i_1, t_1, g^{c f_{i_1,j_1}(t_1)}, y_1), (i_2, t_2, g^{c f_{i_2,j_2}(t_2)}, y_2), \ldots, (i_z, t_z, g^{c f_{i_z,j_z}(t_z)}, y_z)),$$

where y_1 is randomly chosen from G and thought as $m\hat{e}(g^\rho, h_{i_1,j_1})^c$, and $y_k, 2 \le k \le z$, is computed as

$$y_1 \hat{e}(g^\rho, g^c)^{\kappa_{i_k,j_k}} / \hat{e}(g^\rho, g^c)^{\kappa_{i_1,j_1}}.$$

Also, $g^{c f_{i_k,j_k}(t_k)}, 1 \le k \le z$, is computed as $(g^c)^{f_{i_k,j_k}(t_k)}$ since $f_{i_k,j_k}(t_k)$ is known in Step 1.

If some revoked users can together compute the session key m from the challenge with probability ϵ_1, we can compute

$$\begin{aligned} & y_1 \cdot m^{-1} \cdot \hat{e}(g^a, g^c)^{-\kappa_{i_1,j_1}} \\ &= \hat{e}(g^\rho, h_{i_1,j_1})^c \cdot \hat{e}(g, g)^{-ac\kappa_{i_1,j_1}} \\ &= \hat{e}(g^a, g^{b+\kappa_{i_1,j_1}})^c \cdot \hat{e}(g, g)^{-ac\kappa_{i_1,j_1}} = \hat{e}(g, g)^{abc} \end{aligned} \quad (3)$$

with the same probability ϵ_1. This contradicts the BDH assumption.

Let t' be the time for the reduction and solution computation in Equation (3), where t' is polynomially bounded (on the security parameter ω). Thus, if the collusion attack takes $t_1 - t'$, we can solve the BDH problem in time t_1. ■

B. The PkBE-SD-PI Scheme with IND-CCA security

In this subsection, we present a PkBE-SD-PI scheme that achieves IND-CCA security. The idea of construction is the same as the PkBE-PI Π' scheme.

Let $\Phi : \mathcal{N} \times \mathcal{N} \times G \times G_1 \to G_1$ and $\Psi : \mathcal{N} \times \mathcal{N} \times G_1 \to G_1$ be two hash functions, modeled as random oracles. Let $\pi : Z_q \to Z_q$ be a collision-resistant hash function. The scheme is as follows.

- In **Setup**, the system adds Φ, Ψ, π to the public key.

- In **Enc**, for each $S_{i_k,j_k}, 1 \leq k \leq z$, the sender does the following.
  - Compute $A_{i_k,j_k} = \hat{e}(g^\rho, h_{i_k,j_k})^{r_{i_k,j_k}}$, where $r_{i_k,j_k} \in_R Z_q$.
  - Compute $B_{i_k,j_k} = m \oplus \Psi(i_k, j_k, A_{i_k,j_k})$.
  - Compute $C_{i_k,j_k} = \hat{e}(g^\rho, h_{i_k,j_k})^{r_{i_k,j_k}/\pi(B_{i_k,j_k})}$.
  - Compute $D_{0,i_k,j_k} = g^{r_{i_k,j_k}/\pi(B_{i_k,j_k})}$ and $D_{1,i_k,j_k} = g^{r_{i_k,j_k} f_{i_k,j_k}(t_k)/\pi(B_{i_k,j_k})}$.
  - Compute $E_{i_k,j_k} = \Phi(i_k, j_k, D_{0,i_k,j_k}, C_{i_k,j_k}) \oplus B_{i_k,j_k}$.

  The header is

$$Hdr(S,m) = ((i_1, j_1, E_{i_1,j_1}, D_{0,i_1,j_1}, D_{1,i_1,j_1}), \ldots, (i_z, j_z, E_{i_z,j_z}, D_{0,i_z,j_z}, D_{1,i_z,j_z}).$$

- In **Dec**, a user U_l in S finds S_{i_k,j_k} to which he belongs and does the following:
  - Use $D_{0,i_k,j_k}, D_{1,i_k,j_k}$ and his share s_{l,i_k,j_k} to compute $C_{i_k,j_k} = \hat{e}(g^\rho, h_{i_k,j_k})^{r_{i_k,j_k}/\pi(B_{i_k,j_k})}$.
  - Compute $B_{i_k,j_k} = E_{i_k,j_k} \oplus \Phi(i_k, j_k, D_{0,i_k,j_k}, C_{i_k,j_k})$.
  - Compute $A_{i_k,j_k} = C_{i_k,j_k}^{\pi(B_{i_k,j_k})}$.
  - Compute $m = B_{i_k,j_k} \oplus \Psi(i_k, j_k, A_{i_k,j_k})$.

The correctness of the scheme can be checked easily. The security proof is almost the same as that of Theorem 2.

VI. Discussion and Conclusion

In the PkBE-PI and PkBE-SD-PI schemes, we use different h for different polynomials. Nevertheless, for efficiency we can use the same h for all polynomials. We believe that this won't affect the security of the schemes.

The LSD (Layered SD) broadcast encryption scheme is an improvement of the SD broadcast encryption scheme by the trade-off between the header size and the number of private keys held by each user. Our PkBE-SD-PI schemes can be extended to the PkBE-LSD-PI schemes easily.

We have presented two very efficient public key BE schemes. Both of them have low public and private keys. One of them even have a constant decryption time. Our results show that the efficiency of public key BE schemes is comparable to that of private-key BE schemes.

We are interested in reducing the ciphertext size while keeping other complexities low in the future.

References

[1] N. Attrapadung, H. Imai. Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In *Proceedings of Advances in Cryptology - Asiacrypt 05*, Lecture Notes in Computer Science 3788, pp.100-120, Springer, 2005.

[2] D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proceedings of Advances in Cryptology - Eruocrypt 05*, Lecture Notes in Computer Science 3494, pp.440-456, Springer, 2005.

[3] D. Boneh, M. Franklin. An efficient public key traitor tracing scheme. In *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.338-353, Springer, 1999.

[4] D. Boneh, C. Gentry, B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of Advances in Cryptology - Crypto 05*, Lecture Notes in Computer Science 3621, pp.258-275, Springer, 2005.

[5] D. Boneh, B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the ACM Conference on Computer and Communications Security - CCS 06*, pp.211-220, ACM Press, 2006.

[6] X. Boyen. Miniature CCA2 PK encryption: tight security without redudancy. In *Proceedings of Advances in Cryptology - Asiacrypt 07*, Lecture Notes in Computer Science 4833, pp.485-501, Springer, 2007.

[7] Y. Dodis, N. Fazio. Public key broadcast encryption for stateless receivers. In *Proceedings of Digital Right Management 02 - DRM 02*, Lecture Notes in Computer Science 2696, pp.61-80, Springer, 2002.

[8] Y. Dodis, N. Fazio. Public key broadcast encryption secure against adaptive chosen ciphertext attack. In *Proceedings of Public Key Cryptography - PKC 03*, Lecture Notes in Computer Science 2567, pp.100-115, Springer, 2003.

[9] A. Fiat, M. Naor. Broadcast encryption. In *Proceedings of Advances in Cryptology - Crypto 93*, Lecture Notes in Computer Science 773, pp.480-491, Springer, 1993.

[10] D. Galindo. Boneh-Franklin identity based encryption revisited. In Proceedings of ICALP 05, Lecture Notes in Computer Science 3580, pp.791-802, Springer, 2005.

[11] M.T. Goodrich, J.Z. Sun, R. Tamassia. Efficient Tree-Based Revocation in Groups of Low-State Devices. In *Proceedings of Advances in Cryptology - Crypto 04*, Lecture Notes in Computer Science 3152, pp.511-527, Springer, 2004.

[12] D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Proceedings of Advances in Cryptology - Crypto 02*, Lecture Notes in Computer Science 2442, pp.47-60, Springer, 2002.

[13] K. Kurosawa, Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proceedings of Advances in Cryptology - Eurocrypt 98*, Lecture Notes in Computer Science 1403, pp.145-157, Springer, 1998.

[14] K. Kurosawa, T. Yoshida. Linear code implies public-key traitor tracing. In *Proceedings of Public Key Cryptography*, Lecture Notes in Computer Science 2274, pp.172-187, Springer, 2002.

[15] J.W. Lee, Y.H. Hwang, P.J. Lee. Efficient public key broadcast encryption using identifier of receivers. In *Proceedings of International Conference on Information Security Practice and Experience - ISPEC 06*, Lecture Notes in Computer Science 3903, pp.153-164, Springer, 2006.

[16] D. Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of Advances in Cryptology - Crypto 01*, Lecture Notes in Computer Science 2139, pp.41-62, Springer, 2001.

[17] M. Naor, B. Pinkas. Efficient trace and revoke schemes. In *Proceedings of Financial Cryptography 00*, Lecture Notes in Computer Science 1962, pp.1-20, Springer, 2000.

[18] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), pp.612-613, 1979.

[19] W.-G. Tzeng, Z.-J. Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In *Proceedings of Public Key Cryptography - PKC 01*, Lecture Notes in Computer Science 1992, pp.207-224, Springer, 2001.

[20] P. Wang, P. Ning, D.S. Reeves. Storage-efficient stateless group key revocation. In *Proceedings of the 7th Information Security Conference - ISC 04*, Lecture Notes in Computer Science 3225, pp.25-38, Springer, 2005.

[21] E.S. Yoo, N.-S. Jho, J.J. Cheon, M.-H. Kim. Efficient broadcast encryption using multiple interpolation methods. In *Proceedings of ICISC 04*, Lecture Notes in Computer Science 3506, pp.87-103, Springer, 2005.

[22] M. Yoshida, T. Fujiwara. An efficient traitor tracing scheme for broadcast encryption. In *Proceedings of 2000 IEEE International Symposium on Information Theory*, pp.463, IEEE Press, 2000.

Key Establishment Schemes Against Storage-Bounded Adversaries in Wireless Sensor Networks

Shi-Chun Tsai, Wen-Guey Tzeng, Kun-Yi Zhou

***Abstract*—In this paper we re-examine the attacking scenario about wireless sensor networks. It is generally assumed that the adversary picks up all radio communications of sensor nodes without any loss and stores the eavesdropped messages for later use. We suggest that in some situations the adversary may not be able to pick up all radio communications of sensor nodes. Therefore, we propose the storage-bounded adversary model for wireless sensor networks, in which the adversary's storage is bounded.**

We propose two key establishment schemes for establishing shared keys for neighboring sensor nodes in the storage-bounded adversary model. The first scheme needs special beacon nodes for broadcasting random bits. In the second scheme, some sensor nodes play the role of beacon nodes. Our results are theoretical in some sense. Nevertheless, we can adjust them for realistic consideration.

***Index Terms*—Bounded-storage model, key establishment, unconditional security, wireless sensor network.**

I. Introduction

A wireless sensor network usually consists of a large number of small autonomous sensor nodes. Each sensor node has some level of computing power, a limited size of storage, a set of sensors for exploring the environment and a small antenna for communicating with the outside world. One way of deploying a wireless sensor network is to scatter senor nodes in the field randomly. Then, these sensor nodes form a network autonomously via their built-in programs. Due to restriction of small antenna, each sensor node can communicate with its geographic neighbors only. We say that two sensor nodes are *neighbored* if they can communicate with each other via radio directly. In some situations, we may deploy a set of special nodes, called *beacon nodes*, for broadcasting instructions and data to the sensor nodes. A beacon node is more powerful so that its radio signal could cover a larger area.

There are some security issues about wireless sensor networks, such as, communication security, message authentication, node authentication, etc. We are concerned about the key establishment problem, which is to establish a shared (secret) key for two neighboring sensor nodes via the public radio link. The established key is later used for secure communication (encryption) or authentication. The key establishment problem for wireless sensor networks has been studied actively. In this paper we re-examine the attacking scenario about wireless sensor networks. It is generally assumed that the adversary picks up all radio communications of sensor nodes without any loss and stores the eavesdropped messages for later use. We suggest that this may not be the case. For example, the radio quality of a sensor node is not very good and its coverage area is small. It is hard for the adversary to get all communications between sensor nodes. Therefore, we propose the *storage-bounded adversary* model for wireless sensor networks to capture the nature of *incomplete eavesdropping*. In this model, the adversary cannot eavesdrop all communications of the sensor nodes. We could conceptually think that the adversary's storage is limited so that it cannot store all communications. The storage-bounded adversary model has been studied in the cryptographic field for its advanced view. It explores the possibility of encryption in the era of quantum computation. We bring the model to wireless sensor networks for exploring an alternative adversary model.

By considering the storage-bounded adversary, we propose two key establishment schemes. The first scheme needs some special beacon nodes for broadcasting random bits. In the second scheme, some sensor nodes play the role of beacon nodes. Our results are theoretical in some sense. Nevertheless, we can adjust them for realistic consideration.

Our key establishment schemes have the following properties. Firstly, they do not pre-load secrets to sensor nodes. This saves quite a lot of setup work before sensor nodes are deployed to the field. Secondly, the connectivity rate of neighboring sensor nodes is very high and the probability of repeated keys is very low. Thirdly, even if the adversary captures a large fraction of the deployed sensor nodes, almost all of the shared keys of un-compromised links remain secure. We note that most key pre-distribution schemes allow only a small fraction of sensor nodes to be compromised by the adversary. Finally, the shared keys in the first scheme are unconditionally secure. Furthermore, since all shared keys are generated in the field without pre-loaded secrets in sensor nodes, shared keys can be updated from time to time.

We do not consider the adversary that applies other types of attacks, such as node impersonation, node replication, etc. There have been many proposed countermeasures [5]–[7]. If we need them, we can simply use them without too much

The authors are with Computer Science Department, National Chiao Tung University, Hsinchu, Taiwan 30050. Their emails are sctsai@cs.nctu.edu.tw, wgtzeng@cs.nctu.edu.tw, and kyzhou@cs.nctu.edu.tw. The corresponding author is Professor Wen-Guey Tzeng

Research supported in part by projects NSC-96-2628-E-009-011-MY3, NSC-97-2221-E-009-064-MY3 NSC-97-2219-E-009-006 (TWISC), and MoE-97W803.

Manuscript received August 06, 2008; revised October 02, 2008; accepted, November 08, 2008.

The corresponding Associate Editor is Professor Dapeng Wu.

effort.

Related work. Maurer [8] first proposed the storage-bounded adversary model. Cachin and Maurer [2] proposed a complete solution for encryption under the storage-bounded adversary model.

For key pre-distribution, Blom [1] proposed a scheme for multiple parties to establish pairwise keys. Eschenauer and Gligor [6] proposed to assign a random subset of the key space to each sensor node. They showed that two neighboring nodes can establish a shared key from their own key pools with a reasonable probability. Chan, et al. [3], Du, et al. [5], and Liu and Ning [7] improved the basic random key pre-distribution scheme of Eschenauer and Gilgor by using multiple random key pools for each sensor node. Ren, et al. [12] discussed how to pre-distribute keys in large scale.

Miller and Vaidya [9] proposed a key pre-distribution scheme by assuming that the communication channels between sensor nodes use the orthogonal frequency-division multiplexing technology. They considered that these channels cannot be eavesdropped all together. Thus, each sensor node broadcasts its pre-loaded secrets to its neighboring nodes through these channels randomly. Due to the characteristics of the channels, only a part of broadcasted secrets are obtained by the adversary. Then, two neighboring sensor nodes can use the common secrets to establish their shared key. The essence of their assumption is similar to *incomplete eavesdropping*. But, they used it in designing a key pre-distribution scheme. Our schemes are not key pre-distributed. Furthermore, our analysis technique is quite different.

II. Preliminaries

We assume that the sensor nodes are scattered to the field randomly. Each sensor node has no post-deployment knowledge about the other sensor nodes. All it can do is to use its antenna to communicate with its neighboring sensor nodes.

The adversary can eavesdrop all communications of sensor nodes. But, due to storage limitation it can store only a fraction of the eavesdropped messages. After that, the adversary compromises a fraction of the sensor nodes (compromised sensor nodes) and gets the secrets inside them. Then, the adversary tries to infer the shared key held by two neighboring sensor nodes that are not compromised.

Our first key establishment scheme is called *Key Establishment with Beacons in the Storage-Bounded Model*, denoted as KeB-Sb. The beacon nodes are deployed like the sensor nodes, but with a much less number. Each beacon node broadcasts random bits that are received by the sensor nodes within its radio range. Then, two neighboring sensor nodes use the received bits to establish their shared key.

The second key establishment scheme is called *Key Establishment in the Storage-Bounded Model*, denoted as Ke-Sb. Ke-Sb needs no beacon nodes. Each sensor node can play the role of a beacon node. Unlike KeB-Sb, a sensor node that broadcasts random bits establishes shared keys with its neighboring sensor nodes.

The used parameters and notations of the schemes are shown in Table I.

TABLE I
The used parameters and notations.

- n: the number of deployed sensor nodes in a wireless sensor network. Assume that the sensor node set is $\{V_1, V_2, \ldots, V_n\}$.
- α: the number of broadcasted random bits by a beacon node.
- β: the number of stored bits, with respect to each beacon or beaming node, by the adversary.
- γ: the number of broadcasted random bits by a beaming node.
- κ: the length of the shared keys established among neighboring sensor nodes. Typically, κ is 128-bit long.
- $\mu = 2\sqrt{\kappa\alpha}$: the number of randomly stored bits of a sensor node for each beacon node in the KeB-Sb scheme.
- $K_{i,j}$: the shared key computed by sensor node V_i for its neighbor V_j within a bacon or beaming node.
- $p_{complete}$: the probability of forming a complete network.
- H: a cryptographic hash function with κ-bit output.
- G: a pseudorandom generator that stretches a short random bit string to a very long pseudorandom bit string.
- $|S|$: the number of elements in set S.
- $a \ll b$: a is much smaller than b.

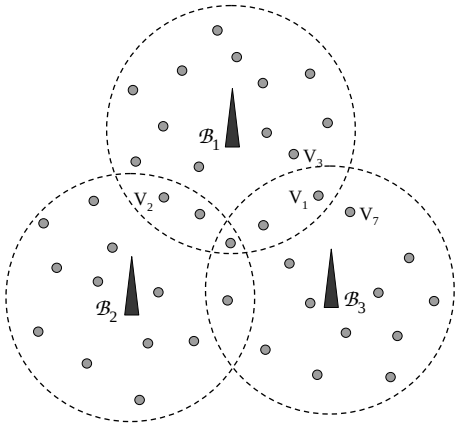


Fig. 1. Deployment of sensor and beacon nodes in a field. Each beacon node uses a different frequency to broadcast random bits and each sensor receives and stores some of them.

In our analysis, we use a Chernoff bound to derive a closed form for approximating security probabilities [11]. Let X_i be identical and independent Boolean random variables with expectation $E(X_i) = \theta$, $1 \le i \le t$. Then, almost all values of $\sum_{i=1}^{t} X_i$ are around its mean $E(\sum_{i=1}^{t} X_i) = t\theta$, that is, for any $0 < \epsilon \le 1$,

$$\Pr[\sum_{i=1}^{t} X_i \ge (1+\epsilon)t\theta] \le e^{-t\theta\epsilon^2/3}.$$

III. Scheme: KeB-Sb

Assume that the field deployment of sensor and beacon nodes is like that in Figure 1, in which a dot is a sensor node and a triangle is a beacon node. We assume that there are z beacon nodes $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_z$. We shall determine an appropriate z later. Without loss of generality, we only present steps for beacon node $\mathcal{B}_1$ and sensor nodes $V_1, V_2, \ldots, V_m$ within its radio range. The adversary gets a fraction of the broadcasted random bits of $\mathcal{B}_1$.

The Scheme. The sensor nodes within $\mathcal{B}_1$ use the steps in Figure 2 to establish their shared keys. Those within other

1) $\mathcal{B}_1$ generates and broadcasts α random bits on the fly.
2) Each V_i, $1 \leq i \leq m$, randomly stores μ bits $r_{i_1} r_{i_2} \cdots r_{i_\mu}$. Let $S_i = \{i_1, i_2, \ldots, i_\mu\}$.
3) Each V_i, $1 \leq i \leq m$, does the following:
   a) Exchange S_i with each of its neighbors V_j via their direct radio link;
   b) Let $S_{i,j} = S_i \cap S_j = \{s_1, s_2, \ldots, s_l\}$. If $|S_{i,j}| = l \geq \kappa$, compute $K_{i,j} = H(r_{s_1} r_{s_2} \cdots r_{s_l})$.
   c) Erase the stored bits $r_{i_1} r_{i_2} \cdots r_{i_\mu}$ from its memory.

Fig. 2. KEB-SB: Steps of establishing shared keys between neighboring sensor nodes within the radio range of the beacon node $\mathcal{B}_1$.

beacon nodes do the same thing. The idea is that $\mathcal{B}_1$ broadcasts α random bits and each sensor node randomly stores μ bits. Then, two neighboring sensor nodes exchange the indices of their stored bits and find their common bits. Finally, they compute the shared key from the common bits by taking the hash value of the common bits. It is easy to check that $K_{i,j} = K_{j,i}$ since V_i and V_j found their common bits from the publicly exchanged indices.

It is critical that some sensor nodes V lie within the radio coverage areas of many beacon nodes, say, $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_\tau$. Assume that $\mathcal{B}_i$'s use different frequencies for broadcasting so that they won't interfere with each other. In this case, V establishes shared keys with its neighboring sensor nodes within various beacon nodes $\mathcal{B}_k$, $1 \leq k \leq \tau$. Thus, a network that connects all sensor nodes can be formed. For example, the sensor node V_1 has a shared key $K_{1,3}$ with V_3 within $\mathcal{B}_1$ and a shared key $K_{1,7}$ with V_7 within $\mathcal{B}_3$. V_1 plays as a connecting node between the sensor nodes within $\mathcal{B}_1$ and the sensor nodes within $\mathcal{B}_3$.

Probability of Establishing Shared Keys. In the scheme each sensor node within a beacon node stores $\mu = 2\sqrt{\kappa\alpha}$ broadcasted bits randomly. Two neighboring sensor nodes within a beacon node will have 4κ common bits on average. Furthermore, the probability that two neighboring sensor nodes have at least κ common bits is $1 - e^{-\kappa/4}$ at least. For $\kappa = 128$, $1 - e^{-\kappa/4} \approx 1$. The following lemma shows this fact, where S and T are the sets of indices of stored bits by two neighboring sensor nodes, respectively.

Lemma 1 ([4]): If S and T are randomly chosen from the $2\sqrt{\kappa\alpha}$-element subsets over $\{1, 2, \ldots, \alpha\}$, then, for sufficiently large α,

$$\Pr_{S,T}[|S \cap T| < \kappa] < e^{-\kappa/4}.$$

Security of Shared Keys. Assume that the adversary stores $\beta = \delta\alpha$ bits of the broadcasted α bits, where $\delta < 1$ is a constant. The security of shared keys depends on δ and κ. Two neighboring sensor nodes within a beacon node have $l = 4\kappa$ common bits on average and the adversary gets a fraction δl of them on average. Although the number l of common stored bits is a random variable, we take the average $l = 4\kappa$ for simplifying analysis. We show that the probability that the adversary gets up to $(\delta + \epsilon)l$ common bits is very low, where $\delta + \epsilon < 1$.

Let $A \subset \{1, 2, \ldots, \alpha\}$ be the set of indices of the stored bits by the adversary, $|A| = \beta$, and B the set of indices of the commonly stored bits by two neighboring sensor nodes, $|B| = l$. We fix A first. The probability that the adversary stores $(\delta + \epsilon)l$ common bits is, for $\delta + \epsilon < 1$ and integer $l(\delta + \epsilon)$,

$$\Pr_B[|A \cap B| \geq (\delta + \epsilon)l] = \sum_{i=(\delta+\epsilon)l}^{l} \frac{\binom{\beta}{i}\binom{\alpha-\beta}{l-i}}{\binom{\alpha}{l}}.$$

It is hard to derive a closed form for the above equation. Nevertheless, we can compute a pretty tight upper bound. In the above computation the elements in B are randomly chosen one by one from $\{1, 2, \ldots, \alpha\}$ *without replacement*. However, if α is much larger than l, we can think that the elements are randomly chosen one by one *with replacement*. Let B' be a multi-set with l elements randomly chosen one by one from $\{1, 2, \ldots, \alpha\}$ *with replacement*. Since α is indeed much large than l in our schemes, we can safely say that

$$\Pr_B[|A \cap B| \geq (\delta + \epsilon)l] \approx \Pr_{B'}[|A \cap B'| \geq (\delta + \epsilon)l],$$

which is bounded by the following lemma.

Lemma 2: Let A be a fixed subset of $\{1, 2, \ldots, \alpha\}$ with $|A| = \beta$ and B', $|B'| = l \ll \beta$, a multi-subset randomly chosen from $\{1, 2, \ldots, \alpha\}$ with replacement. It holds that

$$\Pr_{B'}[|A \cap B'| \geq (\delta + \epsilon)l] \leq e^{-l\epsilon^2/(3\delta)}.$$

Proof: Let X_i be the indicator random variable for whether the ith chosen element of B' is in A, $1 \leq i \leq l$. We have $|A \cap B'| = \sum_{i=1}^{l} X_i$ and $E(\sum_{i=1}^{l} X_i) = \delta l$. Since X_i's are independent, by the Chernoff bound, we have

$$\begin{aligned}
\Pr_{B'}[|A \cap B'| \geq (\delta + \epsilon)l] &= \Pr[\sum_{i=1}^{l} X_i \geq (\delta + \epsilon)l] \\
&= \Pr[\sum_{i=1}^{l} X_i \geq \delta l(1 + \epsilon/\delta)] \leq e^{-\delta l(\epsilon/\delta)^2/3} \\
&= e^{-l\epsilon^2/(3\delta)}.
\end{aligned}$$

■

Since the above holds for any fixed A, the probability holds no matter how the adversary stores broadcasted bits. For $\kappa = 128$, $\delta = 2/3$, $\epsilon = 1/4$, we have

$$\Pr_{B'}[|A \cap B'| \geq (11/12)l] < e^{-16}.$$

In this case, the adversary does not know at least $(1 - \delta - \epsilon)l \approx 43$ common bits of two neighboring sensor nodes within a beacon node.

Probability of Complete Connectivity. We now compute the number of beacon nodes that are needed for high $p_{complete}$. The most important factor for $p_{complete}$ is the size of the overlapping area of radio coverage since the sensor nodes within the overlapping area connect sensor nodes within different beacon nodes. Let R be the radius of the field and r be the radius of the radio coverage of a beacon node. Recall that there are z beacon nodes. We take a very conservative and ideal estimate for the required z. Here, we assume that

each overlapping area is shared by three beacon nodes. For each beacon node, the overlapping area of coverage is at least

$$(\pi r^2 z - \pi R^2)/2z,$$

where $r^2 z - R^2 > 0$. If we want the number of sensor nodes within the overlapping area of a beacon node to be at least c, we need

$$\frac{n}{\pi R^2}(\frac{\pi r^2 z - \pi R^2}{2z}) \geq c,$$

which implies

$$z \geq \frac{nR^2}{nr^2 - 2cR^2} \quad (1)$$

With these c connecting sensor nodes within each beacon node, the probability that the sensor nodes within the beacon node are isolated from the whole network is at most $(2e^{-\kappa/4})^c$.

There are n/z sensor nodes within each beacon node on average. The probability that any one of them fails to connect to another sensor node is at most $(n/z)e^{-\kappa/4}$. Since there are z beacon nodes, the probability $p_{complete}$ that all sensor nodes are connected is at least

$$1 - z((n/z)e^{-\kappa/4} + (2e^{-\kappa/4})^c),$$

which is very close to 1 for a relatively large n, say, $n = 1000$.

Our analysis is based on idealistic assumptions, such as a good frequency management and the coverage of the random deployment is reasonably well. For practical consideration, please see, e.g., [10].

IV. Scheme: Ke-Sb

In the situation that no beacon nodes exist, we let some sensor nodes play the role of broadcasting random bits. We call these sensor nodes as *beaming nodes*. Assume that each sensor node becomes a beaming node with probability p independently, where p will be determined later. The choice of p is to have enough beaming nodes to cover the whole field. A field deployment is shown in Figure 3, in which V_1 to V_9, denoted as triangles, are the beaming nodes. Note that since a beaming node uses a seed to generate pseudorandom bits, the adversary's computing power should be polynomial-time bounded, instead of unboundedness.

The Scheme. The Ke-Sb scheme is shown in Figure 4. A beaming node V_j broadcasts γ pseudorandom bits $G(s_j) = r_{j,1}r_{j,2}\ldots r_{j,\gamma}$ and each sensor node V_i within its radio range stores 4κ bits of them randomly. Then, the sensor node V_i sends the indices $(j, j_1), (j, j_2), \ldots, (j, j_{4\kappa})$ of the stored bits to V_j and computes the shared key $K_{i,j}$ which is the hash value of its stored bits. V_j computes the stored bits of V_i from the random seed s_j and the shared key $K_{j,i}$ in the same way. It is necessary that a beaming node uses a pseudorandom generator to generate pseudorandom bits since these pseudorandom bits are used later for computing shared keys with its neighboring sensor nodes.

Security of Shared Keys. The security analysis of a shared key is the same as that of the KeB-Sb scheme. Recall that an adversary has a storage of β bits. By Lemma 2, the probability that the adversary gets $l(\delta + \epsilon)$ of the stored bits of a sensor node is less than

$$e^{-4\kappa\epsilon^2\gamma/(3\beta)}.$$

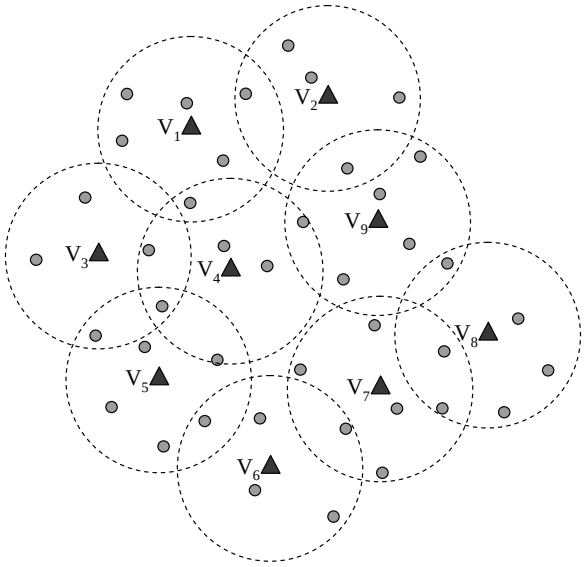


Fig. 3. Deployment of sensor nodes in a field. Some sensor nodes become beaming nodes for broadcasting random bits.

- Each V_i, $1 \leq i \leq n$, randomly acts a beaming node with probability p. Without loss of generality, let $V_1, V_2, \ldots, V_\tau$ be the beaming nodes and $V_{\tau+1}, V_{\tau+2}, \ldots, V_n$ be the non-beaming sensor nodes.

1) Each beaming node $V_j, 1 \leq j \leq \tau$, generates a secret seed s_j randomly and broadcasts γ pseudorandom bits $G(s_j) = r_{j,1}r_{j,2}\cdots r_{j,\gamma}$.
2) Each non-beaming sensor node $V_i, \tau + 1 \leq i \leq n$, does the following. Assume that V_i is within radio range of beaming nodes $V_1, V_2, \ldots, V_\rho$, wlog.
   a) Randomly store 4κ bits $r_{j,j_1}r_{j,j_2}\cdots r_{j,j_{4\kappa}}$ from each $V_j, 1 \leq j \leq \rho$. Let $S_{i,j} = \{(j, j_1), (j, j_2), \ldots, (j, j_{4\kappa})\}$, $1 \leq j \leq \rho$.
   b) Send $S_{i,j}$ to V_j, $1 \leq j \leq \rho$.
   c) Compute the shared key $K_{i,j} = H(r_{j,j_1}r_{j,j_2}\cdots r_{j,j_{4\kappa}})$ with V_j, $1 \leq j \leq \rho$.
3) Each beaming node V_j, $1 \leq j \leq \tau$, computes the shared key $K_{j,i} = H(r_{j,j_1}r_{j,j_2}\cdots r_{j,j_{4\kappa}})$ by $S_{i,j}$ with each of its *neighboring* sensor nodes V_i, where $r_{j,j_1}r_{j,j_2}\cdots r_{j,j_{4\kappa}}$ is re-computed from its random seed s_j.
4) Each beaming node V_j erases its random seed s_j from its memory, $1 \leq j \leq \tau$.

Fig. 4. Ke-Sb: Steps of establishing shared keys between beaming nodes and their neighboring sensor nodes.

Density of Beaming Nodes. The larger p is, the higher $p_{complete}$ is. Nevertheless, we want to have a smaller p so that the expected number np of beaming nodes is as small as possible. Assume that r is the radius of radio range of a beaming node and R is the radius of the deployment field. Note that this r is smaller than that of a beacon node in the KEB-SB scheme. The expected number of beaming nodes is np, which is equivalent to z, the number beacon nodes. By Equation (1), we need

$$z = np \geq \frac{nR^2}{nr^2 - 2cR^2},$$

where c is the expected number of connecting nodes in the overlapping area of two beaming nodes. Thus, we have

$$p \geq \frac{R^2}{nr^2 - 2cR^2}.$$

V. DISCUSSION

Our schemes are designed on an abstract model of wireless sensor networks. Many details are omitted. Comparison between the conventional and storage-bounded adversary model is uncalled-for since their basic assumptions are fundamentally different. Even though our schemes are theoretical, we can use some techniques to improve their performance on energy consumption, storage requirement and computation cost.

1) No re-send: It is possible that a sensor node does not receive some random bits from beacon or beaming nodes. The sensor node can simply ignore a lost bit and continues to wait for the next one. This does not affect its functionality since only a very small fraction of broadcasted bits are stored by each sensor node. Thus, the beacon and beaming nodes can broadcast in a "raw" mode.
2) Sleeping: In our schemes, random bits are broadcasted for a relatively long period of time. But, the sensor nodes do not store all of them. Thus, the sensor nodes can use the random sleeping technique to reduce energy consumption. Each sensor node stays in a state of very low energy consumption for most time and wakes up to receive bits from time to time.
   Furthermore, when a sensor node needs to receive broadcasted random bits from different beacon or beaming nodes in different frequencies, it can switch to a different frequency in each wake-up. Thus, the beacon or beaming nodes can broadcast random bits at different frequencies without worrying about whether their neighboring sensor nodes can receive them simultaneously.
3) Pseudorandomness: In our schemes, all kinds of nodes need some random bits. Beacon and beaming nodes need to generate random bits for broadcasting and sensor nodes need to generate random indices for picking up broadcasted random bits. In fact, pseudorandom bits can replace random bits for better efficiency. A node can sample a short random seed s from the environment and uses the pseudorandom bit generator G to generate pseudorandom bits $G(s)$.
   It should be noted that if we use pseudorandom bits in the scheme, the storage-bounded adversary should be polynomial-time bounded also, instead of computing-unboundedness. This is because a computing-unlimited adversary can search the seed by the eavesdropped pseudorandom bits and the pseudorandom generator G.

In reality, an adversary may jam the media to block the process of key establishment. It is hard for wireless communications to resist this kind of denial of service attacks. Due to sensor nodes' low hardware profile, it is not practical for them to receive the random bits from a satellite. In the above we only discuss how to establish shared keys for the sensor nodes that are within the radio range of beacon and beaming nodes. For others that are neighbored can establish direct link through the path-key finding process.

VI. CONCLUSIONS

We have introduced the storage-bounded adversary model to wireless sensor networks and proposed two key establishment schemes in this model. We are interested in improving efficiency of the schemes for practicability in the future. We are also interested in proposing different kinds of security schemes for wireless sensor networks in this model.

REFERENCES

[1] R. Blom. An optimal clas of symmetric key generation systems, *EUROCRYPT 84*, pp. 335-338, 1984.
[2] C. Cachin, U.M. Maurer. Unconditional security against memory-bounded adversaries, *CRYPTO 97*, pp.292-306, 1997.
[3] H. Chan, A. Perrig, D. Song. Random key predistribution for sensor networks, *IEEE Symposium on Security and Privacy 03*, pp.197-213, 2003.
[4] Y.Z. Ding. Oblivious transfer in the bounded storage model, *CRYPTO 01*, pp.155-170, 2001.
[5] W. Du, J. Deng, Y.S. Han, P. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks, *ACM CCS 03*, pp.42-51, 2003.
[6] L. Eschenauer, V.D. Gilgor. A key-mamnagement scheme for distributed sensor networks, *ACM CCS 02*, pp.41-47, 2002.
[7] D. Liu, P. Ning. Eatablishing pairwise keys in distributed sensor networks, *ACM CCS 03*, pp.52-61, 2003.
[8] U.M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology* 5(1), pp.53-66, 1992.
[9] M.J. Miller, N.H. Vaidya. Leveraging channel diversity for key establishment in wireless sensor networks, *IEEE INFOCOM 06*, pp.1-12, 2006
[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, B. Srivastava. Coverage problems in wireless ad-hoc sensor networks, *IEEE INFOCOM 01*, pp.1380-1387, 2001.
[11] M. Mitzenmacher, E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. University of Cambridge Press, 2005.
[12] K. Ren, K. Zeng, W. Lou. A new approach for random key predistribution in large scale wireless sensor networks. *Wireless Communications and Mobile Computing* 6(3), pp.307-318, 2006.