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：沈宣佐、陳毅睿、羅日宏、周桂伊

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　99　年　10　月　31 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)。

第一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 2008 的 PKC 會議上發表。第二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。

第二年度，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，結果發表在 IEEE Trans.Wireless Communications 上。

第三年度我們對 permutation code(PC)做了詳細的研究，提出許多 PC 碼的性質及好的編解碼方法，結果發表在 IEEE Trans. Information Theory 上。

關鍵詞：廣播加密、公開金鑰、排序碼、金鑰建立。

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 2008, we 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 2009, 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 2009, we also spent time on the problem of key establishment problem in the wireless sensor networks. We explore a very novel 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 and published in IEEE Trans. Wireless Communications.

In 2010, we studied the error correcting coding problem for permutation codes. We found many interesting properties and proposed efficient decoding algorithms. The result has been accepted by IEEE Trans. Information Theory.

Keywords: Broadcast encryption, public key system, permutation code, key establishment.

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

第一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 2008 的 PKC 會議上發表。第二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。

第二年度，我們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。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。結果發表在 IEEE Trans.Wireless Communications 上。

第三年度我們對 permutation code(PC)做了詳細的研究，提出許多 PC 碼的性質及好的編解碼方法，結果發表在 IEEE Trans. Information Theory 上。

三、計畫成果自評

本計劃我們發表了四篇高水準的會議及期刊論文，還有一篇在審稿中，以成果來看，我們達成了計劃的預定目標。

附件

1. Y.-R. Liu, W.-G. Tzeng. Public key broadcast encryption with low number of keys and constant decryption time. In *Proceedings of International Workshop on Practice and Theory in Public-Key Cryptography (PKC 08)*, LNCS 4939, pp.380-396, 2008.
2. S.-C. Tsai, W.-G. Tzeng, Kun-Yi Zhou. "Key Establishment Schemes Against Storage-Bounded Adversaries in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications* 8(3), pp.1218-1222, 2009.
3. Yi-Ruei Chen and Wen-Guey Tzeng. A Public-Key Traitor Tracing Scheme with an Optimal Transmission Rate, *In Proceedings of the 11th International Conference on Information and Communications Security (ICICS 09)*, LNCS 5927, pp.121-134, 2009.
4. T. Kløve, T.-T. Lin, S.-C. Tsai, W.-G. Tzeng. "Permutation Arrays Under the Chebyshev Distance," *IEEE Transactions on Information Theory* 56(6), pp.2611-2617, 2010.

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

Yi-Ru Liu and Wen-Guey Tzeng

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan 30050
wgtzeng@cs.nctu.edu.tw

Abstract. In this paper we propose three public key BE schemes that have efficient complexity measures. The first scheme, called the BE-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 PK-SD-PI scheme, has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys per user. They are the same as those of the SD scheme. Nevertheless, the 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. The third scheme, called, the PK-LSD-PI scheme, is constructed in the same way, but based on the LSD method. It has $O(r/\epsilon)$ ciphertext size and $O(\log^{1+\epsilon} N)$ private keys per user, where $0 < \epsilon < 1$. The decryption time is also $O(1)$.

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

Keywords: Broadcast encryption, polynomial interpolation, collusion.

1 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

* Research supported in part by NSC projects 96-2628-E-009-011-MY3, 96-3114-P-001-002-Y (iCAST), and 96-2219-E-009-013 (TWISC).

R. Cramer (Ed.): PKC 2008, LNCS 4939, pp. 380–396, 2008.

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 $\langle Hdr(S,m),\ E_m(M)\rangle$, where m is a session key for encrypting M via a symmetric encryption method 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".

1.1 Our Contribution

We propose three public key BE schemes that have efficient complexity measures. The first scheme, called the BE-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 PK-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].

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

Table 1. Comparison of some fully collusion-resistant public key BE schemes

	header size	public-key size	private-key size	decryption cost[♮]
PK-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: BE-PI	$O(r)$	$O(1)$	$O(\log N)$	$O(r)$
Our work: PK-SD-PI	$O(r)$	$O(1)$	$O(\log^2 N)$	$O(1)$
Our work: PK-LSD-PI	$O(r/\epsilon)$	$O(1)$	$O(\log^{1+\epsilon} N)$	$O(1)$

N - the number of users.
r - the number of revoked users.
[†] - the transformed SD scheme [6] instantiated with constant-size HIBE [2].
[‡] - the parallel extension of [17,19,22].
[♭] - the public keys are needed for decrypting the header by a user.
[§] - $N = C^D$.
[♮] - group operation/modular exponentiation and excluding the time for scanning the header.

The PK-SD-PI scheme has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys per user. They are the same as those of the SD scheme. Nevertheless, the decryption 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. The third scheme, called the PK-LSD-PI scheme, is constructed in the same way, but based on the LSD method. It has $O(r/\epsilon)$ ciphertext size and $O(\log^{1+\epsilon} N)$ private keys per user, where $0 < \epsilon < 1$. The decryption time is also $O(1)$.

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 indistinguishably security against adaptive chosen ciphertext attacks. The comparison with some other public key BE schemes with full collusion resistance is shown in Table 1.

1.2 Related Work

Fiat and Naor [8] 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 [7,13,14,17,19,22] 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 [6] transformed the SD and LSD schemes [12,16] into public key SD and LSD schemes, shorted as PK-SD and PK-LSD. 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 PK-SD scheme has $O(r)$ header size, $O(1)$ public keys and $O(\log^2 N)$ private keys. The PK-LSD 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 PK-SD and PK-LSD 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.

2 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, $\hat{e}(g, g) = g_1 \neq 1$ 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 time bound t, there is some k_0 such that for any $k \geq k_0$,

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

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

- Setup(1^z, ID, $\mathcal{U}$). Wlog, let $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$. It takes as input the security parameter z, 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.

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^z, 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, $1 \leq i \leq n$, 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, $n+1 \leq i \leq q_D$, 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 $\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(z)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

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

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 $Adv^{ind\text{-}cca}_{\mathcal{A},\Pi}(z) < \epsilon$.*

In this paper we first give schemes with one-way security against chosen plaintext attacks (OW-CPA security) and then transform them to have IND-CCA security via the Fujisaki-Okamoto transformation [9]. 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^z, 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 $\mathrm{Adv}^{ow\text{-}cpa}_{\mathcal{A},\Pi}(z)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

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

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 $Adv^{ow\text{-}cpa}_{\mathcal{A},\Pi}(z) < \epsilon$.*

3 The BE-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 achieve $O(r)$ ciphertexts, we use $\log N$ polynomials, each for a range of the number of revoked users.

1. **Setup**$(1^z, \text{ID}, \mathcal{U})$: z 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 $g^{a_j^{(i)}} = H_2(\text{ID}\|i\|j)$ for $0 \le i \le \log N$ and $0 \le j \le 2^i$.
     Remark. The underlying polynomials are, $0 \le i \le \log N$,

$$f_i(x) = \sum_{j=0}^{2^i} a_j^{(i)} x^j \pmod q.$$

     The system dealer does not know the coefficients $a_j^{(i)} = \lg H_2(\text{ID}\|i\|j)$. But, this does not matter.
   - Randomly choose a secret $\rho \in Z_q$ and compute g^ρ.
   - Publish the public key $PK = (\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^\rho)$.
   - 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. **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, $1 \le t \le L$,

$$g^{r f_\alpha(i_t)} = (\prod_{j=0}^{L} H_2(\text{ID}\|\alpha\|j)^{i_t^j})^r.$$

   - The ciphertext header $Hdr(S, m)$ is

$$(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (i_1, g^{r f_\alpha(i_1)}), (i_2, g^{r f_\alpha(i_2)}), \ldots, (i_L, g^{r f_\alpha(i_L)})).$$

   - The ciphertext body is $C = E_m(M)$.

3. **Dec**$(SK_k, Hdr(S,m), C)$: $U_k \in S$. The user U_k does the following.
   - Compute $b_0 = \hat{e}(g^r, g^{r_{k,\alpha} f_\alpha(k)}) = g_1^{r r_{k,\alpha} f_\alpha(k)}$.
   - Compute $b_j = \hat{e}(g^{r_{k,\alpha}}, g^{r f_\alpha(i_j)}) = g_1^{r r_{k,\alpha} f_\alpha(i_j)}$, $1 \le j \le L$.
   - Use the Lagrange interpolation method to compute

$$g_1^{r r_{k,\alpha} f_\alpha(0)} = \prod_{j=0}^{L} b_j^{\lambda_j}, \tag{1}$$

     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

$$\frac{m\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}}{\hat{e}(g^r, g^{r_{k,\alpha} f_\alpha(0)} h_\alpha^\rho)} = \frac{m\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}}{\hat{e}(g^r, h_\alpha^\rho) \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}} = m. \tag{2}$$

   - 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).

3.1 Performance Analysis

For each system, the public key is $(\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^\rho)$, which is of size $O(1)$. Since all systems can use the same $(H, E, G, G_1, \hat{e}, g)$, 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 re-arrangement of Equation (1) as

$$\prod_{j=0}^{L} b_j^{\lambda_j} = b_0^{\lambda_0} \cdot \hat{e}(g^{r_{k,\alpha}}, \prod_{j=1}^{L} (g^{r f_\alpha(i_j)})^{\lambda_j}),$$

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

$$\tilde{\lambda}_j = \frac{(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j-i_1)\cdots(i_j-i_{j-1})(i_j-i_{j+1})\cdots(i_j-i_L)} \bmod q, 1 \le j \le L.$$

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.

3.2 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 BE-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 $f_i(x) = \sum_{j=0}^{L} a_j^{(i)} x^j$ 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 $\tau, \kappa, \mu_1, \mu_2, \ldots, \mu_L, w_1, w_2, \ldots, w_L \in Z_q$.
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 $(\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^a)$.
   (d) The following is implicitly computed.
   - Set $f_\alpha(i) = w_i, 1 \le i \le L$.
   - Let $g^{a_0^{(\alpha)}} = g^{f_\alpha(0)} = g^a \cdot g^\tau = g^{a+\tau}$.
   - Compute $g^{a_i^{(\alpha)}}$, $1 \le i \le L$, from $g^{a_0^{(\alpha)}}$ and $g^{f_\alpha(j)} = g^{w_j}, 1 \le j \le L$, by the Lagrange interpolation method over exponents.
   - Set $h_\alpha = g^b \cdot g^\kappa = g^{b+\kappa}$.
   - 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.
3. Set the secret keys $(g^{r_{i,j}}, g^{r_{i,j} f_j(i)}, g^{r_{i,j} f_j(0)} h_j^\rho)$, $0 \le j \le \log N$, 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 $g^{r_{i,\alpha} f_\alpha(0)} h_\alpha^\rho = g^{(-b+\mu_i)(a+\tau)} (g^{b+\kappa})^a = g^{a(\mu_i+\kappa) - b\tau + \mu_i \tau}$.
   (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 $g^{r_{i,j} f_j(0)} h_j^\rho = g^{r_{i,j} f_j(0)} (g^a)^{z_j}$.
4. Set the header $(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (1, g^{r f_\alpha(1)}), (2, g^{r f_\alpha(2)}), \ldots, (L, g^{r f_\alpha(L)}))$ as follows:
   (a) Let $g^r = g^c$.
   (b) Compute $g^{r f_\alpha(i)} = (g^c)^{w_i}, 1 \le i \le L$.
   (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 $g^{a_j^{(i)}}$ 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 values in G.

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, $g^{a_i^{(\alpha)}}, 0 \leq i \leq L$ 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 $\kappa, \mu_1, \mu_2, \ldots, \mu_L$ 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 $\epsilon_1 = \epsilon$ by computing the following:

$$\begin{aligned} y \cdot m^{-1} \cdot \hat{e}(g^a, g^c)^{-\kappa} &= \hat{e}(g^\rho, h_\alpha)^r \cdot \hat{e}(g,g)^{-ac\kappa} \\ &= \hat{e}(g^a, g^{b+\kappa})^c \cdot \hat{e}(g,g)^{-ac\kappa} \\ &= \hat{e}(g,g)^{abc}. \end{aligned} \tag{3}$$

Let t' be the time for this reduction and the solution computation in Equation (3). 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.

4 The BE-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. There are some standard techniques of transforming OW-CPA security to IND-CCA security. Here we present such a scheme Π' based on the technique in [9].

The IND-CCA security of the Fujisaki-Okamoto transformation depends only on the OW-CPA security of the public key encryption scheme, the FG security of a symmetric encryption scheme $\mathcal{E}$, and the γ-uniformity of the public key encryption scheme. The FG-security is the counterpart of the IND-security for symmetric encryption. A public key encryption scheme is γ-uniform if for every key pair (pk, sk), every message x, and $y \in \{0,1\}^*$, $\Pr[E_{pk}(x) = y] \leq \gamma$. Before applying the transformation, we check the following things:

1. The transformation applies to public key encryption, while ours is public key broadcast encryption. Nevertheless, if the authorized set S is fixed, our public

key broadcast encryption scheme is a public key encryption scheme with public key $pk = (PK, S)$. In the definition of IND-CCA security (Definition 1), the adversary $\mathcal{A}$ selects a target set S^* of users to attack in the **Init** stage and S^* is fixed through the rest of the attack. Thus, we can discuss the attack of $\mathcal{A}$ with a fixed target set S^*. Note that $\mathcal{A}$ is a static adversary.

2. Let S be a fixed authorized set of users. For every m and every $y \in \{0,1\}^*$, $\Pr[Hdr(S,m) = y]$ is either 0 or $1/q \simeq 1/2^z$, where z is the security parameter (the public key size). Thus, our broadcast encryption scheme is 2^{-z}-uniform if the authorized set is fixed.

Let $\mathcal{E} : K \times G_1 \to G_1$ be a symmetric encryption scheme with FG-security, where K is the key space of $\mathcal{E}$. Let $H_3 : G_1 \times G_1 \to Z_q$ and $H_4 : G_1 \to K$ be two hash functions. The modification of Π for Π' is as follows.

- In the **Setup** algorithm, add $\mathcal{E}, H_3, H_4$ to PK.
- In the **Enc** algorithm,

$$Hdr(S,m) = (g^r, \sigma\hat{e}(g^\rho, h_\alpha)^r, \mathcal{E}_{H_4(\sigma)}(m), \\ (i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), \ldots, (i_L, g^{rf_\alpha(i_L)})),$$

  where σ is randomly chosen from G_1 and $r = H_3(\sigma, m)$.
- In the **Dec** algorithm, we first compute $\bar{\sigma}$ as described in the BE-PI scheme. Then, we compute the session key $\bar{m}$ from $\mathcal{E}_{H_4(\sigma)}(m)$ by using $\bar{\sigma}$. We check whether $\sigma\hat{e}(g^\rho, h_\alpha)^r = \bar{\sigma}\hat{e}(g^\rho, h_\alpha)^{H_3(\bar{\sigma},\bar{m})}$ and $g^{rf_\alpha(i_j)} = g^{f_\alpha(i_j)H_3(\bar{\sigma},\bar{m})}$, $1 \le j \le L$. If they are all equal, $\bar{m}$ is outputted. Otherwise, $\perp$ is outputted.

Let q_{H_3}, q_{H_4} and q_D be the numbers of queries to H_3, H_4 and the decryption oracles, respectively. Our scheme Π' is IND-CCA-secure.

Theorem 2. *Assume that the BDH problem is (t_1, ϵ_1)-hard and the symmetric encryption $\mathcal{E}$ is (t_2, ϵ_2) FG-secure. The scheme Π' is $(t, \epsilon, q_{H_3}, q_{H_4}, q_D)$-IND-CCA secure in the random oracle model, where t' is some polynomially bounded time,*

$$t = \min\{t_1 - t', t_2\} - O(2z(q_{H_3} + q_{H_4})) \ \textit{and}$$
$$\epsilon = (1 + 2(q_{H_3} + q_{H_4})\epsilon_1 + \epsilon_2)(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-z+1})^{-q_D} - 1.$$

This theorem is proved by showing that if Π' is not IND-CCA-secure, then either Π is not OW-CPA-secure or $\mathcal{E}$ is not FG-secure directly. The OW-CPA security of Π is based on the BDH assumption. We note that the application of the transformation to other types of schemes could be delicate. Galindo [10] pointed out such a case. Nevertheless, the problem occurs in the proof and is fixable without changing the transformation or the assumption. The detailed proof will be given in the full version of the paper.

5 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

the subsets in $\mathcal{C}$, that is, $S = \cup_{i=1}^{w} S_w$, where $S_i \in \mathcal{C}$ are disjoint, $1 \leq i \leq w$. Each subset S_i in $\mathcal{C}$ is associated with a private key k_i. A user is assigned a set of keys such that he can derive the private keys of the subsets to which he belongs. The subset keys k_i cannot be independent. Otherwise, each user may hold too many keys. It is preferable that the subset keys have some relations, for example, one can be derived from another. Thus, each user U_k is given a set SK_k of keys so that he can derive the private key of a subset to which he belongs. A subset-cover based broadcast encryption scheme plays the art of choosing a collection $\mathcal{C}$ of subsets, assigning subset and user keys, and finding subset covers.

5.1 The PK-SD-PI Scheme

We now present our PK-SD-PI scheme, which is constructed by using the polynomial interpolation technique on the collection of subsets in [16]. The system setup is similar to that of the BE-PI scheme. Consider a complete binary tree T of $\log N + 1$ levels. The nodes in T are numbered differently. 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". For each subtree T_i of η levels (level 1 to level η from top to bottom), we define the degree-1 polynomials

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

where $a_{j,0}^{(i)} = \lg H_2(\text{ID}\|i\|j\|0)$ and $a_{j,1}^{(i)} = \lg H_2(\text{ID}\|i\|j\|1)$, $2 \leq j \leq \eta$. For a user U_k in the subtree T_i of η levels, he is given the private keys

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

for $2 \leq j \leq \eta$, where 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. 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. Here, we use h^{ρ} in all private keys in order to save space in the header.

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

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

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 $f_j^{(i)}(t)$, where node t is in the j-th level of subtree T_i, only 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 $f_j^{(i)}(s)$, for some $s \neq t$. In Figure 1, the share $f_3^{(i)}(t)$ 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 subset cover for $\mathcal{U} \backslash R$, the header is

$$(m\hat{e}(g^{\rho}, h)^r, g^r, (i_1, t_1, g^{r f_{j_1}^{(i_1)}(t_1)}), \ldots, (i_z, t_z, g^{r f_{j_z}^{(i_z)}(t_z)})),$$

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

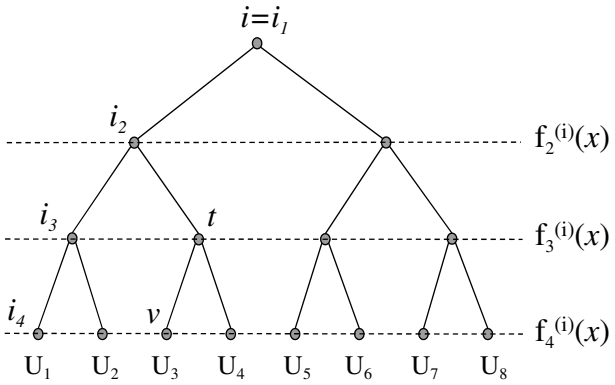

Fig. 1. Level polynomials, private keys and broadcasted shares for subtree T_i

For decryption, a non-revoked user finds $i_k, t_k, g^{rf_{j_k}^{(i_k)}(t_k)}$ (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 BE-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.

Theorem 3. *Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PK-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 PK-SD-PI scheme is similar to that for the BE-PI scheme. In the PK-SD-PI scheme, all polynomials $f_j^{(i)}(x)$ 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 $S^* = \mathcal{U} \backslash R$. 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 $f(x) = f_\beta^{(\alpha)}(x) = a_1 x + a_0 \pmod q$. Wlog, let $\{U_1, U_2, \ldots, U_l\}$ be the set of revoked users that have the secret share about $f(t)$. The reduction to the BDH problem is as follows. Recall that the public key of the PK-SD-PI method is $(\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^\rho)$.

1. Let g be the generator in the system and $g^\rho = g^a$.
2. Set $f(t) = w$ and compute $g^{f(t)} = g^w$, where w is randomly chosen from Z_q.
3. Let $g^{a_0} = g^{f(0)} = g^a \cdot g^\tau$, where τ is randomly chosen from Z_q.
4. Compute g^{a_1} from $g^{f(t)}$ and g^{a_0} via the Lagrange interpolation.
5. 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.
6. Set $h = g^b \cdot g^\kappa$, where κ is randomly chosen from Z_q.
7. The $f(x)$-related secret share of $U_i, 1 \le i \le l$, is computed as $(g^{r_i}, g^{r_i f(t)}, g^{r_i f(0)} h^\rho)$, where $g^{r_i} = g^{-b} \cdot g^{\mu_i}$ and μ_i is randomly chosen from Z_q. Note that $g^{r_i f(0)} h^\rho = g^{a(\mu_i+\kappa)-b\tau+\mu_i\tau}$ can be computed from the setting in the previous steps.
8. The non-$f(x)$-related secret shares of $U_i, 1 \le i \le l$, can be set as follows. Let f' be a polynomial related to subtree α' and level β', where t' is in the β'-th level and $U_i \in S_{\alpha',t'}$. The secret share $(g^{r_i'}, g^{r_i' f'(t')}, g^{r_i' f'(0)} h^\rho)$ of U_i is computed from $(g^{r_i}, g^{r_i f(t)}, g^{r_i f(0)} h^\rho)$. Let $f'(t') = w'$, $f'(0) = f(0) + a'$ and $r_i' = r_i + r'$, where w', a', and r' are randomly chosen from Z_q. Thus, $g^{r_i'} = g^{r_i} \cdot g^{r'}$, $g^{r_i' f'(t')} = (g^{r_i'})^{w'}$ and $g^{r_i' f'(0)} h^\rho = (g^{r_i f(0)} h^\rho) \cdot g^{r' f(0)} \cdot g^{r_i a'} \cdot g^{r' a'}$. Note that the hash values $H_2(\text{ID}\|\alpha'\|\beta'\|0)$ and $H_2(\text{ID}\|\alpha'\|\beta'\|1)$ can be answered accordingly.
9. Set the challenge as

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

where y is randomly chosen from G and thought as $m\hat{e}(g^\rho, h)^c$. Note that $g^{c f_{j_k}^{(i_k)}(t_k)}, 1 \le k \le z$, can be computed since $f_{j_k}^{(i_k)}(t_k)$ is a number randomly chosen from Z_q, as described in Step 2.

If the revoked users $U_1, U_2, \ldots, U_l$ can together compute the session key m from the challenge with probability ϵ_1, we can compute

$$\begin{aligned} y \cdot m^{-1} \cdot \hat{e}(g^a, g^c)^{-\kappa} &= \hat{e}(g^\rho, h)^c \cdot \hat{e}(g, g)^{-ac\kappa} \\ &= \hat{e}(g^a, g^{b+\kappa})^c \cdot \hat{e}(g, g)^{-ac\kappa} = \hat{e}(g, g)^{abc} \end{aligned} \tag{4}$$

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

Let t' be the time for the reduction and solution computation in Equation (4), where t' is polynomially bounded. Thus, if the collusion attack takes $t_1 - t'$, we can solve the BDH problem in time t_1.

Similarly, we can modify our PK-SD-PI scheme to have IND-CCA security like Section 4

5.2 The PK-LSD-PI Scheme

The LSD method is an improvement of the SD method by using a sub-collection $\mathcal{C}'$ of $\mathcal{C}$ in the SD method. The basic observation is that $S_{i,t}$ can be decomposed to $S_{i,k} \cup S_{k,t}$. The LSD method delicately selects C' such that each $S_{i,t} \in \mathcal{C}$ is either in $\mathcal{C}'$ or equal to $S_{i,k} \cup S_{k,t}$, where $S_{i,k}$ and $S_{k,t}$ are in $\mathcal{C}'$. The subset cover found for $\mathcal{U} \backslash R$ in the SD method is used except that each $S_{i,t}$ in the cover, but not in C', is replaced by two subsets $S_{i,k}$ and $S_{k,t}$ in C'. Thus, each user belongs to a less number of $S_{i,t}$'s in C' such that it holds a less number of private keys.

We consider the basic case of the LSD method, in which each user holds $(\log n)^{3/2}$ private keys. There are $\sqrt{\log n}$ "special" levels in T. The root is at a special level and every level of depth $k \cdot \sqrt{\log n}$, $1 \leq k \leq \sqrt{\log n}$, is special. A layer is the set of the levels between two adjacent special levels. Each layer has $\sqrt{\log n}$ levels. The collection $\mathcal{C}'$ of the LSD method is

$$\{S_{i,t} : \text{ nodes } i \text{ and } t \text{ are in the same layer, or node } i \text{ is at a special level}\}.$$

There are two types of $S_{i,t}$'s in $\mathcal{C}'$. The first type is that node i is in a special level and the second type is that nodes i and t are in the same layer. Every non-revoked set $\mathcal{U} \backslash R$ can be covered by at most $4|R| - 2$ disjoint subsets in $\mathcal{C}'$.

Our PK-LSD-PI scheme is as follows. Since $\mathcal{C}'$ is just a sub-collection of $\mathcal{C}$ in the SD method, our PK-LSD-PI scheme is almost the same as the PK-SD-PI scheme except that some polynomials for type-2 $S_{i,t} \in \mathcal{C}'$ are unnecessary. Consider a user U_k (or its corresponding leaf node). For his ancestor node i at a special layer (type-1 $S_{i,t}$'s), U_k is given the private keys (corresponding to subtree T_i) by the same way as the PK-SD-PI method. There are $\sqrt{\log n}$ such i's and each T_i has at most $\log n$ levels. In this case, U_k holds $(\log n)^{3/2}$ private keys. For his ancestor node i and nodes t in the same layer (type-2 $S_{i,t}$'s), choose degree-1 polynomials for the levels between i and its (underneath) adjacent special level only. There are at most $\sqrt{\log n}$ such polynomials and U_k is assigned corresponding $\sqrt{\log n}$ private keys as the PK-SD-PI scheme does. In this case, U_k holds at most $\log n \cdot \sqrt{\log n}$ private keys since U_k has $\log n$ ancestors. Overall, each user U_k holds at most $2(\log n)^{3/2}$ private keys.

Security. We show that the scheme described in this subsection is one-way secure.

Theorem 4. *Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PK-LSD-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 collection of $S_{i,t}$'s for covering $\mathcal{U} \backslash R$ in the LSD method is a sub-collection of that in the SD method. The way of assigning private keys to users is the same as that of the PK-SD-PI scheme except that we omit the polynomials that are never used due to the way of choosing a subset cover in the LSD method. In the random oracle model, we can simply consider a particular $S_{\alpha,t}$ in the subset cover for $\mathcal{U} \backslash R$. Since all conditions are the same, the rest of proof is the same as that in Theorem 3.

With the same extension in [12], we can have a PK-LSD-PI scheme that has $O(1)$ public keys and $O(\log^{1+\epsilon})$ private keys, for any constant $0 < \epsilon < 1$. The header size is $O(r/\epsilon)$, which is $O(r)$ for a constant ϵ. The decryption cost excluding the time of scanning the header is again $O(1)$.

6 Conclusion

We have presented very efficient public key BE schemes. They have low public and private keys. Two 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.

Acknowledgement

We thank Eike Kiltz and Michel Abdalla for valuable comments on the manuscript.

References

1. Attrapadung, N., Imai, H.: Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 100–120. Springer, Heidelberg (2005)
2. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
3. Boneh, D., Franklin, M.: An efficient public key traitor tracing scheme. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999)
4. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
5. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Proceedings of the ACM Conference on Computer and Communications Security - CCS 2006, pp. 211–220. ACM Press, New York (2006)
6. Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
7. Dodis, Y., Fazio, N.: Public key broadcast encryption secure against adaptive chosen ciphertext attack. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2002)
8. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
9. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)

10. Galindo, D.: Boneh-Franklin identity based encryption revisited. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 791–802. Springer, Heidelberg (2005)
11. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient Tree-Based Revocation in Groups of Low-State Devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
12. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
13. Kurosawa, K., Desmedt, Y.: Optimum traitor tracing and asymmetric schemes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 145–157. Springer, Heidelberg (1998)
14. Kurosawa, K., Yoshida, T.: Linear code implies public-key traitor tracing. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 172–187. Springer, Heidelberg (2002)
15. Lee, J.W., Hwang, Y.H., Lee, P.J.: Efficient public key broadcast encryption using identifier of receivers. In: Chen, K., Deng, R., Lai, X., Zhou, J. (eds.) ISPEC 2006. LNCS, vol. 3903, pp. 153–164. Springer, Heidelberg (2006)
16. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
17. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
18. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
19. Tzeng, W.-G., Tzeng, Z.-J.: A public-key traitor tracing scheme with revocation using dynamic shares. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 207–224. Springer, Heidelberg (2001)
20. Wang, P., Ning, P., Reeves, D.S.: Storage-efficient stateless group key revocation. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 25–38. Springer, Heidelberg (2004)
21. Yoo, E.S., Jho, N.-S., Cheon, J.J., Kim, M.-H.: Efficient broadcast encryption using multiple interpolation methods. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 87–103. Springer, Heidelberg (2005)
22. Yoshida, M., Fujiwara, T.: An efficient traitor tracing scheme for broadcast encryption. In: Proceedings of 2000 IEEE International Symposium on Information Theory, p. 463. IEEE Press, Los Alamitos (2000)

A Public-Key Traitor Tracing Scheme with an Optimal Transmission Rate*

Yi-Ruei Chen and Wen-Guey Tzeng

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan 30010, ROC
yrchen.cs98g@nctu.edu.tw, wgtzeng@cs.nctu.edu.tw

Abstract. The way of transmitting the encrypted digital content to the legitimate subscribers over a broadcast channel has wide commercial applications, such as Pay-TV, DVD, etc. In order to discourage the legitimate subscribers from giving away their decryption keys, the traitor tracing scheme comes up. In this paper, we propose a public-key traitor tracing scheme that has optimal transmission rate. In other words, our scheme enables everyone to transmit the encrypted digital contents almost without any redundancy. As for tracing, our scheme supports black-box tracing, i.e., identifying colluders without opening the pirate decoder. Moreover, in our scheme, the storage requirement for legitimate subscribers and digital content broadcasters is smaller than that of previous schemes.

Keywords: Traitor tracing, transmission rate, fingerprinting code.

1 Introduction

Consider the scenario that a data supplier distributes the digital content over a broadcast channel. The data supplier gives a secret key to each legitimate subscriber. Then the data supplier broadcasts the encrypted digital content and the legitimate subscribers decrypt the digital content by their secret keys. The protection for some Pay-TV, CD-ROM, DVD, and online databases is based on this scenario. However, some malicious subscribers (called *traitors*) might give copies of their secret keys to illegitimate users (called *pirates*). Then the pirates decrypt the digital content for free. In order to solve the problem above, the *traitor tracing* scheme comes up.

The goal of traitor tracing schemes is to discourage legitimate subscribers from giving away their secret keys. One approach is to give each subscriber a unique set of secret keys that both decrypt the encrypted digital content and identify ("trace") the subscribers. To avoid being traced by a tracer, the traitors may collude to obfuscate their secret keys and generate a new secret key set (called

* The research was supported in part by projects NSC 96-2628-E-009-011-MY3 and 98-2219-E-009-003-

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 121–134, 2009.

pirate key). We call a traitor tracing scheme *t-collusion resistant* if at least one of the traitors can be identified when t traitors collude to generate a pirate key in this way. If t is the number of all legitimate users, the traitor tracing scheme is called *fully-collusion resistant*. Note that the traitors may embed the pirate keys into a "tamper-resistant" hardware (called *pirate decoder*) to prevent the tracer from reading any data inside. So, during the tracing, the tracer has to treat the pirate decoder as a *black box* – only the outcome of a pirate decoder can be examined.

In many previous traitor tracing schemes, the overhead of broadcasting the encrypted digital content is proportional to the number of legitimate subscribers. But in some applications, such as Pay-TV, the number of legitimate subscribers might be up to millions. This is a great burden. The approach of *public-key traitor tracing* schemes is to enable everyone (e.g. Pay-TV stations) to broadcast the encrypted digital content. The *public traceability* of a traitor tracing scheme allows everyone with a pirate decoder to trace the traitors. In order to measure the efficiency of traitor tracing schemes, we consider the "transmission rate" of encrypted digital content ("ciphertext"), that is, the ratio of the size of ciphertext to the size of the digital content. We also care about the storage requirements of subscribers' secret keys, and the broadcast keys.

Related work. The *traitor tracing* scheme was first introduced by Chor, Fiat and Naor [9], and later refined in [15,10]. The concept of *public-key traitor tracing* schemes was proposed in Kurosawa and Desmedt [14], and Boneh and Franklin [2]. The traitor tracing schemes in [2,3,8,12,11,14,13,16,18,21,22] are public-key traitor tracing schemes. In [8], Chabanne, Phan, and Pointcheval proposed the concept of *public traceability*. A class of traitor tracing schemes relying on the usage of *fingerprinting codes* [5,20] was introduced by Kiayias and Yung [13]. They showed that if the plaintexts are large (e.g. multimedia content), it is possible to obtain constant transmission rate. For example, the schemes in [8,18,17,11] have constant transmission rate. While considering the transmission rate, we have two main categories in the traitor tracing schemes:

- Schemes with *no constant transmission rate* [2,4]: These schemes are well-suited to encrypt small digital content (usually using for the session-key exchanges in the "hybrid encryption"). The user-key size and the public-key size are often relatively small in these schemes. But the transmission rate in these schemes is often linear or sublinear to the maximal number of colluders.
- Schemes with *constant transmission rate* [13,8,11] (including ours): These schemes are well-suited to encrypt large digital content (e.g. multimedia content). They are all constructed by using the fingerprinting codes. One advantage of these schemes is that they often have efficient black-box tracing algorithms. Nevertheless, the user-key size and the public-key size are often relatively large (according to the codeword length in the fingerprinting codes).

Our Contributions. We propose a public-key traitor tracing schemes with efficient black-box tracing and the optimal transmission rate. The storage

Table 1. Scheme Comparison

	transmission rate	user-key size	public-key size	black-box tracing	traceability
BF99 [2]	$2t+1$	$2t$	$2t+1$	inefficient	private
BSW06 [4]	$6\sqrt{N}$	1	$4\sqrt{N}+2$	O	public
KY02 [13]	~ 3	2ℓ	4ℓ	O	private
CPP05 [8]	~ 1	2ℓ	$\ell+1$	X	private
FNP07 [11]	~ 1	2ℓ	10ℓ	O	private
Ours	~ 1	$\ell+2$	$2\ell+1$	O	private

† ℓ: the codeword length in fingerprinting code
† N: the total number of legitimate subscribers

requirements in our scheme for user-keys and pubic-keys are smaller than previous schemes that have the constant (or optimal) transmission rate. Our scheme is based on a fingerprinting code, and an all-or-nothing transformation [6,7,19]. The idea is to encrypt a block of the output of an all-or-nothing transformation by a special public-key scheme. The encryption does not entail much overhead and allows us to feed indistinguishable messages for tracing. The comparison with other related schemes is given in Table 1. We show that our scheme is semantically secure based on the DDH assumption and the indistinguishability of PKE-AONT. We also show that our traitor tracing scheme is t-collusion resistant under the DDH assumption.

2 Preliminaries

Notations. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c \in \mathbb{N}$, there exists an integer $k_0 \in \mathbb{N}$ such that $f(k) \leq k^{-c}$ for all $k \geq k_0$, denoted by $neg(k)$. We use $x \stackrel{\$}{\leftarrow} X$ to denote that x is chosen from the set X uniformly. Let $\mathcal{M}$ be the plaintext space.

Fingerprinting Codes. The fingerprinting technique with fingerprinting codes embeds a specific fingerprint (*codeword*) to each document copy so that one can identify which copy of document by examining the embedded fingerprint. The codeword is a collection of some alphabets. The traitors will collude and try to modify their codewords to avoid being identified. However, the coalition of the traitors is restricted by the *marking assumption*: the traitors are only able to compare their codewords and make a modification from their respective codewords differing in some positions. Under the marking assumption, the possible modified codeword set from t traitor's codewords set W is called a *feasible set* of W.

- For a codeword $w \in \{0,1\}^\ell$, we write $w = w_1 w_2 ... w_\ell$, where $w_i \in \{0,1\}$.
- Let $W = \{w^{(1)}, ..., w^{(t)}\} \subseteq \{0,1\}^\ell$. We say that a codeword $\bar{w}$ is *feasible* for W if $\forall i \in \{1,2,...\ell\}$ $\exists j \in \{1,2,...,t\}$ s.t. $\bar{w}_i = w_i^{(j)}$. For example, if $W = \{0101, 1111\}$, the codewords $\{0101, 0111, 1101, 1111\}$ are feasible for W.
- For a codeword set $W \subseteq \{0,1\}^\ell$, we say that the feasible set of W, denoted by $F(W)$, is the set of all codewords that are feasible for W.

Table 2. Length of the fingerprinting codes with respect to the number N of codewords

	t-collusion resistant	fully-collusion resistant
BS98 [5]	$\ell = O(t^4 \log(N/\epsilon) \log(1/\epsilon))$	$\ell = O(N^3 \log(N/\epsilon))$
T03 [20]	$\ell = O(t^2 \log(N/\epsilon))$	$\ell = O(N^2 \log(N/\epsilon))$

A fingerprinting code scheme consists of two probabilistic polynomial-time algorithms: the *codeword generation* algorithm G and the *codeword tracing* algorithm T. Algorithm G generates codeword set $\Gamma = \{w^{(1)}, ..., w^{(N)}\} \in (\{0,1\}^\ell)^N$ for some $\ell > 0$ and the trace-key tk. By taking a pirate codeword $\bar{w}$ and tk as input, algorithm T outputs at least one of the traitors who collude to generate $\bar{w}$. The fingerprinting codes enable a data supplier to distribute an "object" with many fingerprinting copies. Assume $\Gamma \in (\{0,1\}^\ell)^N$ is a codeword set generated by the algorithm G, and denote a L-length object to be distributed by P. First we partition the object P into ℓ blocks and embed exactly one "mark" in each block. Let $P_{j,s}$ be the j-th block which contains the j-th mark with state $s \in \{0,1\}$. For each block, choose a random key $K_{j,s}$, the distributed data will be $\{f_{K_{j,s}}(P_{j,s}) | 1 \leq j \leq \ell, s \in \{0,1\}\}$, where f is an encryption scheme. Let $w^{(u)} \in \Gamma$ be the codeword in Γ and associated with user u. The private user-key for u is $\{K_{1,w_1^{(u)}}, K_{2,w_2^{(u)}}, ..., K_{\ell,w_\ell^{(u)}}\}$, and $P(w^{(u)}) = P_{1,w_1^{(u)}} || P_{2,w_2^{(u)}} || ... || P_{\ell,w_\ell^{(u)}}$ will be the copy of P implied by $w^{(u)}$.

Boneh and Shaw [5] constructed a fully-collusion resistant fingerprinting code as well as t-collusion resistant secure codes. Tardos [20] proposed a shorter code. The comparison of their codeword lengths is in Table 2, where N is the number of codewords, and ϵ is the security parameter.

All-Or-Nothing Transformation. An all-or-nothing transformation (AONT) Σ is an efficient, unkeyed, and randomized transformation with the property that it is hard to invert unless the entire output is known [19]. Σ maps an ℓ'-block sequence x, together with a random string ρ to an ℓ-block sequence y with the following properties:

- Given x and ρ, $y \xleftarrow{\$} \Sigma(x; \rho)$ can be computed efficiently.
- Given all blocks of y, $x \leftarrow \Sigma^{-1}(y)$ can be computed efficiently.
- It is infeasible to get any information about x if any block of y is missing.

Notice that the AONT expands plaintext size by roughly $1 + 1/\ell$. This results in an asymptotical unitary ciphertext-to-plaintext ratio.

Decisional Deffie-Hellman (DDH) Assumption. For a cyclic group $\mathbb{G}$ with a generator g. Let $\mathcal{V}$ be the distribution $\{(g, g^u, g^v, g^{uv})\}$ and $\mathcal{R}$ be the distribution $\{(g, g^u, g^v, g^w)\}$. For any polynomial time adversary $\mathcal{A}$, $\mathcal{A}$ distingishes the two distributions $\mathcal{V}$ and $\mathcal{R}$ with negligible function of λ, i.e., $|\Pr[\mathcal{A}(X) = 1 : X \in \mathcal{V}] - \Pr[\mathcal{A}(X) = 1 : X \in \mathcal{R}]| = neg(|\mathbb{G}|)$.

3 The Notion for Public-Key Traitor Tracing Scheme

A public-key traitor tracing scheme is a 4-tuple of probabilistic polynomial-time algorithms $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Trace})$, where

- $\mathsf{Setup}(1^\lambda, N)$. The setup takes as input a security parameter λ and N, the number of users in the system. The algorithm outputs a public broadcast-key BK, a secret trace-key TK, and the private user-key SK_u for each legitimate subscriber u.
- $\mathsf{Encrypt}(\mathrm{BK}, M)$. The encryption algorithm takes as input the public broadcast-key BK and a message $M \in \mathcal{M}$. The algorithm outputs a ciphertext C.
- $\mathsf{Decrypt}(\mathrm{SK}_u, C)$. The decryption algorithm takes as input the private user-key SK_u of user u and a ciphertext C. The algorithm outputs a message M or $\perp$.
- $\mathsf{Trace}^{\mathcal{D}}(\mathrm{TK})$. The tracing algorithm takes as input the private trace-key TK and queries the pirate decoder $\mathcal{D}$ as a black-box oracle. The algorithm outputs a traitor set S which is a subset of $\{1, ..., N\}$.

Moreover, the scheme must satisfy the correctness property as follows:

For all $u \in \{1, ..., N\}$ and for all $M \in \mathcal{M}$: if $\langle \mathrm{BK}, \mathrm{TK}, (\mathrm{SK}_1, ..., \mathrm{SK}_N) \rangle \xleftarrow{\$} \mathsf{Setup}(1^\lambda, N)$ and $C \xleftarrow{\$} \mathsf{Encrypt}(\mathrm{BK}, M)$, then $\mathsf{Decrypt}(\mathrm{SK}_u, C) = M$.

Semantic Security Game

- *Setup.* The challenger runs Setup, and gives BK to the adversary.
- *Challenge.* The adversary chooses two plaintexts $M_0, M_1 \in \mathcal{M}$ to the challenger. Then the challenger flips a coin $b \in \{0, 1\}$, and gives a ciphertext $C_b \xleftarrow{\$} \mathsf{Encrypt}(\mathrm{BK}, M_b)$ to the adversary.
- *Guess.* The adversary returns a guess $b' \in \{0, 1\}$ of b to the challenger.

The advantage of winning this game by the adversary is $\mathsf{Adv}^{\mathsf{TTS}}_{\mathsf{SS}} := |\Pr[b' = b] - \frac{1}{2}|$

Definition 1 (Semantically secure). *An N-user public-key traitor tracing scheme is semantically secure if for all polynomial time adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{TTS}}_{\mathsf{SS}}$ is a negligible function of the security parameter.*

Traceable against t-collusion Game

- *Setup.* The challenger runs Setup and gives BK to the adversary. The adversary chooses a traitor set $\mathsf{T} = \{u_1, ..., u_t\} \subseteq \{1, ..., N\}$ to the challenger. Then the challenger gives the adversary $\mathrm{SK}_{u_1}, ..., \mathrm{SK}_{u_t}$ to produce a pirate decoder $\mathcal{D}$.
- *Trace.* By taking a pirate decoder $\mathcal{D}$ as a decryption oracle, the challenger runs the algorithm $\mathsf{Trace}^{\mathcal{D}}(\mathrm{TK})$ to obtain a traitor set $\mathsf{S} \subseteq \{1, ..., N\}$.

The adversary wins this game if (1) $\mathcal{D}$ decrypts all valid ciphertext with a constant probability δ, i.e., $\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathrm{BK}, M)) = M] \geq \delta$, and (2) $\mathsf{S} \cap \mathsf{T} \neq \emptyset$.

The probability of adversary winning this game is $\mathsf{Adv}^{\mathsf{TTS}}_{\mathsf{TR}}$.

Definition 2 (Traceable against t-collusion). *An N-user pubic-key traitor tracing scheme is traceable against t-collusion if for all polynomial time adversaries $\mathcal{A}$ of corrupting t users and any constant $\delta > 0$, $\mathsf{Adv}_{\mathsf{TR}}^{\mathsf{TTS}}$ is a negligible function of the security parameter.*

4 A Public-Key Traitor Tracing Scheme for Two Users

In this section, we give a construction of a public-key traitor tracing scheme for two users. Then we show that this scheme is semantically secure and traceable against 1-collusion. Indeed, this scheme will be a subscheme in the construction of our public-key traitor tracing scheme for N users in section 5.

4.1 Our Construction

Our public-key traitor tracing scheme for two users is 2-PK-TTS = (2-Setup, 2-Encrypt, 2-Decrypt, 2-Trace), where

2-Setup(1^λ) Given a security parameter λ, the algorithm generates a λ-bit prime q, a cyclic group $\mathbb{G}$ of order q, and a generator g of $\mathbb{G}$. Then the algorithm chooses $f(x) = a_0 + a_1 x \pmod q$, where $a_0, a_1 \xleftarrow{\$} \mathbb{Z}_q^*$ and sets
- Public broadcast-key $bk := \langle g, (g^{a_0}, g^{a_1})\rangle$
- Secret trace-key $tk := \langle f(x)\rangle$
- User-key $sk_\sigma := \langle i_\sigma, f(i_\sigma)\rangle$, where $i_\sigma \in \mathbb{Z}_q^*$, $\forall \sigma \in \{0,1\}$

2-Encrypt(bk, m) Given bk and a plaintext $m \in \mathcal{M}$, the algorithm chooses $r, j \xleftarrow{\$} \mathbb{Z}_q^*$, where $j \neq i_0$ or i_1, computes $g^{rf(j)} = (g^{a_0}(g^{a_1})^j)^r$ and outputs the ciphertext $c := \langle mg^{ra_0}, g^r, (j, g^{rf(j)})\rangle$.

2-Decrypt(sk_σ, c) Given a ciphertext $c = \langle A, R, (j, W)\rangle$ and a user-key sk_σ, the algorithm computes the plaintext based on the lagrange interpolation: $m = A/W^{\frac{-i_\sigma}{j-i_\sigma}} R^{f(i_\sigma)\frac{-j}{i_\sigma - j}}$.

2-Trace$^{\mathcal{D}}(tk)$ Given a pirate decoder $\mathcal{D}$ that decrypts all valid ciphertext perfectly as a decryption oracle. The algorithm does:
1. 2-TrEncrypt(bk, m) The algorithm chooses $r, \hat{r}, j \xleftarrow{\$} \mathbb{Z}_q^*$ and computes a probe ciphertext $\hat{c} \xleftarrow{\$} \langle A = mg^{ra_0}, R = g^r, (j, \hat{W} = g^{\hat{r}f(j)})\rangle$.
2. $\forall \sigma \in \{0,1\}$, pre-compute $V_\sigma = \hat{W}^{\frac{-i_\sigma}{j-i_\sigma}} R^{f(i_\sigma)\frac{-j}{i_\sigma - j}}$.
3. $\forall \sigma \in \{0,1\}$, if $\mathcal{D}(\hat{c}) = A/V_\sigma$, output $\mathsf{S} = \{\sigma\}$; else output $\mathsf{S} = \{0,1\}$.

4.2 Security Analysis of our 2-PK-TTS scheme

Theorem 1. *The* 2-PK-TTS *scheme is semantically secure under the DDH assumption.*

Proof. By contradiction, assume that there exists a 2-PK-TTS scheme adversary $\mathcal{A}$ that wins the semantic security game with a non-negligible advantage $\epsilon > 0$. We construct an algorithm $\mathcal{B}$ that breaks the DDH assumption with non-negligible advantage ϵ as follows:

- *Setup.* Algorithm $\mathcal{B}$ is given as input an instance (g, g^u, g^v, X) of the DDH assumption, and it wants to determine whether $X = g^{uv}$ or X is a random element in $\mathbb{G}$ ($\mathbb{G}$ has prime order q). $\mathcal{B}$ chooses $a_0, a_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ and sets $bk = \langle g, (g^u, g^{a_1})\rangle$ to $\mathcal{A}$. (we see that $f(x) = u + a_1 x \pmod q$)
- *Challenge.* $\mathcal{A}$ chooses two plaintexts $m_0, m_1 \in \mathcal{M}$ to $\mathcal{B}$, then $\mathcal{B}$ flips a coin $b \in \{0, 1\}$, and sets the challenge $c_b = \langle m_b X, g^v, (j, X(g^v)^{a_1 j})\rangle$ to $\mathcal{A}$, where $j \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$.
- *Guess.* $\mathcal{A}$ outputs $b' \in \{0, 1\}$ to $\mathcal{B}$. If $b' = b$, $\mathcal{B}$ answers that $X = g^{uv}$; else $\mathcal{B}$ answers that X is a random element in $\mathbb{G}$.

If $X = g^{uv}$, $\mathcal{A}$ gets a valid ciphertext $c_b = \langle m_b g^{uv}, g^v, (j, g^{v(u+a_1 j)})\rangle$. Therefore, $\mathcal{A}$ answers $b' = b$ successfully with probability $\frac{1}{2} + \epsilon$;

If X is a radom element in $\mathbb{G}$, $\mathcal{A}$ gets an invalid ciphertext. In this case, $\mathcal{A}$ answers $b' = b$ successfully with probability $\frac{1}{2}$.

Hence, $\mathcal{B}$ solves the DDH problem with non-negligible advantage ϵ. This is a contradiction to the DDH assumption. So we conclude that such adversary $\mathcal{A}$ does not exist.

Theorem 2. *The* 2-PK-TTS *scheme is traceable against 1-collusion under the DDH assumption.*

Proof. By contradiction, assume that there exists an adversary $\mathcal{A}$ that, given the public-key bk and one of user-keys sk_σ in 2-PK-TTS scheme, $\mathcal{A}$ produces a pirate decoder $\mathcal{D}$ that decrypts all valid ciphertexts perfectly, i.e., $\Pr[\mathcal{D}(2\text{-}\mathsf{Encrypt}(bk, m)) = m : \mathcal{D} \stackrel{\$}{\leftarrow} \mathcal{A}(bk, sk_\sigma), \sigma \in \{0, 1\}] = 1$. But when given a probe ciphertext $\hat{c}$, $\mathcal{D}$ outputs a different value from the pre-computed values in 2-Trace algorithm with non-negligible probabilistic $\epsilon > 0$, i.e., $\Pr[\mathcal{D}(\hat{c}) \neq A/V_\sigma] = \epsilon$. We construct an algorithm $\mathcal{B}$ that breaks the DDH assumption with non-negligible advantage $\frac{\epsilon}{2}$ as follows:

- *Setup.* Algorithm $\mathcal{B}$ is given as input an instance (g, g^u, g^v, X) of DDH assumption, and it wants to determine whether $X = g^{uv}$ or X is a random element in $\mathbb{G}$. $\mathcal{B}$ chooses $i, z \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ and gives $\mathcal{A}$ $bk = \langle g, (g^u, g^{a_1} = (\frac{g^z}{g^u})^{i^{-1}})\rangle$. $\mathcal{A}$ chooses a traitor set $\mathsf{T} = \{0\}$ or $\{1\}$ to $\mathcal{B}$. Then $\mathcal{B}$ gives $\mathcal{A}$ $sk = \langle i, z\rangle$ to produces a pirate decoder $\mathcal{D}$.
- *Trace.* By taking a pirate decoder $\mathcal{D}$ as a decryption oracle, $\mathcal{B}$ runs the modified 2-Trace as follows:
  1. Choose $A \stackrel{\$}{\leftarrow} \mathbb{G}$, and $j \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, where $j \neq i$. Compute $W = X(\frac{(g^v)^z}{X})^{ji^{-1}}$ and set the ciphertext as $\bar{c} \leftarrow \langle A, g^v, (j, W)\rangle$.
  2. Pre-compute $V \leftarrow W^{\frac{-i}{j-i}}(g^v)^{z\frac{-j}{i-j}}$.
  3. If $\mathcal{D}(\bar{c}) = A/V$, $\mathcal{B}$ answers that $X = g^{uv}$ or X is a random element in $\mathbb{G}$ randomly; else $\mathcal{B}$ answers that X is a random element in $\mathbb{G}$.

If $X = g^{uv}$, ciphertext $\bar{c}$ is a valid ciphertext, since

$$X(\frac{(g^v)^z}{X})^{ji^{-1}} = g^{uv}(\frac{(g^v)^z}{g^{uv}})^{ji^{-1}} = g^{uv}((\frac{g^z}{g^u})^{i^{-1}})^{vj} = g^{uv}(g^{a_1})^{vj} = g^{v(u+a_1 j)}.$$

In this case, $\mathcal{D}(\bar{c}) = A/V$, $\mathcal{B}$ gives the correct answer with probability $\frac{1}{2}$;

If X is a random element in $\mathbb{G}$, ciphertext $\bar{c}$ is an invalid ciphertext. In this case, $\mathcal{D}(\bar{c}) \neq A/V$ with probability ϵ, and $\mathcal{D}(\bar{c}) = A/V$ with probability $1-\epsilon$. Therefore, $\mathcal{B}$ gives the correct answer with probability $\epsilon + \frac{1}{2}(1-\epsilon) = \frac{1}{2} + \frac{\epsilon}{2}$.

Hence, $\mathcal{B}$ solves the DDH problem with non-negligible advantage $\frac{\epsilon}{2}$. This is a contradiction to the DDH assumption. So we conclude that such adversary $\mathcal{A}$ does not exist.

5 A Public-Key Traitor Tracing Scheme for N Users

Our construction is based on the use of our 2-PK-TTS scheme, an AONT Σ and a fingerprinting code $\Gamma = \{w^{(1)}, ..., w^{(N)}\}$ over $\{0,1\}^{\ell}$. At a high level, the idea is to "concatenate" ℓ instance of 2-PK-TTS scheme according to the code Γ. Each legitimate subscriber $u \in \{1, ..., N\}$ is associated to a codeword $w^{(u)}$ in Γ and assigned a private user-key set $\mathrm{SK}_u = \{sk_{1,w_1^{(u)}}, ..., sk_{\ell,w_\ell^{(u)}}\}$, where $w_j^{(u)}$ is the j-th bit of the codeword $w^{(u)}$, and $sk_{j,0}, sk_{j,1}$ are the keys for the j-th instance of the 2-PK-TTS scheme. For example, let $\ell = 3$. If the codeword corresponding to legitimate subscriber u is 011, its user-key set is $\{sk_{1,0}, sk_{2,1}, sk_{3,1}\}$. For tracing, we use the j-th 2-PK-TTS to identify the j-th symbol of the pirate codeword, for all $j \in \{1, 2, ..., \ell\}$. Finally, by the tracing algorithm in the fingerprinting code, we find the collusion codeword set for constructing a pirate codeword, i.e., we find the collusion traitor set.

In order to achieve the optimal transmission rate, we notice the cryptosystem PKE-AONT proposed by Zhang, Hanaoka, and Imai [23]. It encrypts some bits of the output of an AONT by a public-key encryption scheme. Given an AONT Σ and a public-key encryption scheme $(\mathsf{G}, \mathsf{E}, \mathsf{D})$, PKE-AONT first transforms the original message M' into an all-or-nothing message $M = m_1||...||m_\ell$ by Σ and then randomly chooses a block of M to encrypt it as $C = m_1||...||m_{k-1}||\mathsf{E}(pk, m_k)||m_{k+1}||...||m_\ell$. For decrypting the ciphertext C, the decryption algorithm first decrypts the k-th block to recover M and compute the original message M' by Σ.

5.1 Our Construction

For convenience, we introduce some notations in our scheme:

- Let Σ be an AONT that maps an ℓ'-block sequence together with a random string to an ℓ-block sequence.
- $\mathsf{MINUS}_k(M)$ Given an $\ell\lambda$-bit message $M = m_1||...||m_\ell$ and a position index $k \in \{1, ..., \ell\}$, the algorithm "minus" the k-th block of M, i.e., $\mathsf{MINUS}_k(M) = m_1||...||m_{k-1}||m_{k+1}||...||m_\ell$.
- $\mathsf{COMB}_k(Y, m)$ Given an $(\ell-1)\lambda$-bit message $Y = y_1||...||y_{\ell-1}$, a λ-bit message m and a position index $k \in \{1, ..., \ell-1\}$, the algorithm first splits Y into $X_1||X_2$, where X_1 is the front $(k-1)\lambda$ bits of Y and X_2 is the rest bits of Y. The algorithm "combines" and outputs the messages with order X_1, m, and X_2, i.e. $\mathsf{COMB}_k(Y, m) = y_1||...||y_{k-1}||m||y_k||...||y_{\ell-1}$.

Our traitor tracing scheme for N users $\Pi = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Trace})$ is as follows:

$\mathsf{Setup}(1^\lambda, N)$. Given a security parameter λ and user number N, the algorithm generates a fingerprinting code $\Gamma = \{w^{(1)}, ..., w^{(N)}\} \in (\{0,1\}^\ell)^N$ for some ℓ. Then it runs 2-Setup ℓ times to generate the keys $\langle (bk_i, tk_i, (sk_{0,i}, sk_{1,i}))_{i=1}^N \rangle$ (but use the same $q, \mathbb{G}, g, i_0, i_1$) and sets

- Public broadcast-key $\mathrm{BK} := \langle g, (g^{a_{0,j}}, g^{a_{1,j}})_{j=1}^\ell \rangle$
  (we denote the k-th key of BK by $\mathrm{BK}_k = (g, (g^{a_{0,k}}, g^{a_{1,k}}))$)
- Secret trace-key $\mathrm{TK} := \langle (f_j(x))_{j=1}^\ell \rangle$
- User-key $\mathrm{SK}_u := \langle w^{(u)}, i_0, i_1, (f_j(i_{w_j^{(u)}})_{j=1}^\ell \rangle$, $\forall u \in \{1, 2, ..., N\}$
  (we denote k-th key of SK_u by $\mathrm{SK}_{u,k} = (i_{w_k^{(u)}}, f_k(i_{w_k^{(u)}}))$)

$\mathsf{Encrypt}(\mathrm{BK}, M')$. Given BK and a plaintext $M' \in \mathcal{M}^{\ell'}$, the algorithm chooses a random string $\rho \xleftarrow{\$} \{0,1\}^\tau$, and computes $\Sigma(M'; \rho) = M = m_1 || ... || m_\ell$. Then it chooses a position index $k \xleftarrow{\$} \{1, 2, ..., \ell\}$, and computes the ciphertext $C \xleftarrow{\$} \langle k, \text{2-}\mathsf{Encrypt}(\mathrm{BK}_k, m_k), \mathsf{MINUS}_k(M) \rangle$.

$\mathsf{Decrypt}(\mathrm{SK}_u, C)$. Given a ciphertext $C = \langle k, c_k, Y \rangle$, user u computes $m_k \leftarrow$ 2-$\mathsf{Decrypt}(\mathrm{SK}_{u,k}, c_k)$ and $M' = \Sigma^{-1}(\mathsf{COMB}_k(Y, m_k))$.

$\mathsf{Trace}^{\mathcal{D}}(\mathrm{TK})$. Given a pirate decoder $\mathcal{D}$ that decrypts all valid ciphertext perfectly as a decryption oracle. The algorithm does:

- For each position index $k \in \{1, 2, ..., \ell\}$,
  1. Compute $\Sigma(M'; \rho) = M = m_1 || m_2 || ... || m_\ell$, where $M' \xleftarrow{\$} \mathcal{M}^{\ell'}$ and $\rho \xleftarrow{\$} \{0,1\}^\tau$.
  2. Call 2-$\mathsf{TrEncrypt}(\mathrm{BK}_k, m_k) \xrightarrow{\$} \hat{c}_k = \langle A_k = m_k g^{r a_{0,k}}, R = g^r, (j, \hat{W}_k = g^{\hat{r} f_k(j)}) \rangle$. Set the probe ciphertext as $\hat{C} \xleftarrow{\$} \langle k, \hat{c}_k, Y = \mathsf{MINUS}_k(M) \rangle$.
  3. $\forall \sigma \in \{0,1\}$, pre-compute $M_{k,\sigma} = \mathsf{COMB}_k(Y, A_k / \hat{W}_k^{\frac{-i_\sigma}{j - i_\sigma}} R^{f_k(i_\sigma) \frac{-j}{i_\sigma - j}})$.
  4. $\forall \sigma \in \{0,1\}$, if $\Sigma(\mathcal{D}(\hat{C}); \rho) = M_{k,\sigma}$, set $w_k^* = \sigma$; else set $w_k^* = 0$ for convenience.
- Recover $w^* = w_1^* w_2^* ... w_\ell^*$, then call the tracing algorithm in fingerprinting code by taking w^* as the input to obtain collude codewords. Finally, output the corresponding traitor set S.

5.2 Security Analysis of Our Scheme

Theorem 3. *The scheme Π is semantically secure under the semantic security of* 2-PK-TTS *and the indistingushability of* PKE-AONT.

Proof. For each position index $k \in \{1, 2, ..., \ell\}$, we use two games to bound the advantage of semantically secure in Π with $\mathsf{Adv}_{\mathsf{SS}}^{\text{2-PK-TTS}}$ and $\mathsf{Adv}_{\mathsf{ind}}^{\text{PKE-AONT}}$ as follows:

Game $\mathbf{G}_0$. Define $\mathbf{G}_0$ as the original semantic security game and let S_0 be the event where $b' = b$, i.e., $\mathsf{Adv}_{\mathsf{SS}}^{\Pi} := |\Pr[S_0] - \frac{1}{2}|$.

Game $\mathbf{G}_1$. This game is identical to $\mathbf{G}_0$ except that in the Encrypt, rather than being set $A_k = m_k g^{r a_{0,k}}$, in $\mathbf{G}_1$ A_k is a random λ-bit string, i.e., $C \xleftarrow{\$} \langle A_k \xleftarrow{\$} \{0,1\}^\lambda, R = g^r, (j, W_k = g^{r f_k(j)})\rangle$, and we let S_1 be the event that $b' = b$ in this game.

Claim: $|\Pr[S_0] - \Pr[S_1]| \leq 2\mathsf{Adv}_{\mathsf{SS}}^{\text{2-PK-TTS}}$.
By reduction, if there exists an adversary $\mathcal{A}$ that distinguishes the challenge of $\mathbf{G}_0$ and $\mathbf{G}_1$ with non-negligible probability $\epsilon > 0$, we use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for breaking the semantic security of 2-PK-TTS scheme with non-negligible advantage as follows:

- *Setup.* Algorithm $\mathcal{B}$ is given as input an instance $bk = \langle g, (g^{a_0}, g^{a_1})\rangle$ of 2-PK-TTS scheme and wants to determine whether the challenge C is construct by $\mathbf{G}_0$ or $\mathbf{G}_1$. $\mathcal{B}$ chooses $a_{0,j}, a_{1,j} \xleftarrow{\$} \mathbb{Z}_q^*$, $\forall j \in \{1, 2, ..., \ell\}\backslash\{k\}$, lets $g^{a_{0,k}} = g^{a_0}, g^{a_{1,k}} = g^{a_1}$, and sets $\mathrm{BK} = \langle g, (g^{a_{o,j}}, g^{a_{1,j}})_{j=1}^{\ell}\rangle$ to $\mathcal{A}$.
- *Challenge.* $\mathcal{A}$ chooses two plaintexts $M_0, M_1 \in \mathcal{M}^{\ell'}$ to $\mathcal{B}$, then $\mathcal{B}$ flips a coin $b' \in \{0,1\}$, and it calls $\Sigma(M_{b'}; \rho) = m_{b',1}||m_{b',2}||...||m_{b',\ell}$, lets $m_{1-b',k} \xleftarrow{\$} \{0,1\}^\lambda$, and sends $m_{b'} = m_{b',k}, m_{1-b'} = m_{1-b',k}$ to 2-PK-TTS scheme challenger. Then 2-PK-TTS scheme challenger flips a coin $b \in \{0,1\}$ and sets the challenge $c_b \xleftarrow{\$} \text{2-Encrypt}(bk, m_b)$ to $\mathcal{B}$. Finally, $\mathcal{B}$ sends $\mathcal{A}$ the challenge $C_{b'} = \langle k, c_b, Y = m_{b',1}||...||m_{b',k-1}||m_{b',k+1}||...||m_{b',\ell}\rangle$.
- *Guess.* $\mathcal{A}$ outputs $\hat{b} \in \{0,1\}$ to $\mathcal{B}$. Then $\mathcal{B}$ gives $\hat{b}$ as its guess to 2-PK-TTS scheme challenger.

By the above construction, we see that $\mathcal{B}$ "interpolates" between $\mathbf{G}_0$ and $\mathbf{G}_1$ for $\mathcal{A}$:

- If $b' = b$, $\mathcal{A}$ gets a challenge in $\mathbf{G}_0$;
- If $b' = 1 - b$, $\mathcal{A}$ gets a challenge in $\mathbf{G}_1$.

Thus, it holds that $\Pr[S_0] = \Pr[\hat{b} = b' | b' = b]$ and $\Pr[S_1] = \Pr[\hat{b} = b' | b' = 1 - b]$, and we get

$$\begin{aligned}
\Pr[\hat{b} = b] &= \Pr[\hat{b} = b | b' = b] \Pr[b' = b] + \Pr[\hat{b} = b | b' = 1 - b] \Pr[b' = 1 - b] \\
&= \frac{1}{2}(\Pr[\hat{b} = b | b' = b] + \Pr[\hat{b} = b | b' = 1 - b]) \\
&= \frac{1}{2}(\Pr[\hat{b} = b | b' = b] + 1 - \Pr[\hat{b} = 1 - b | b' = 1 - b]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[\hat{b} = b' | b' = b] - \Pr[\hat{b} = b' | b' = 1 - b]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[S_0] - \Pr[S_1]).
\end{aligned}$$

It follows that $|\Pr[S_0] - \Pr[S_1]| = 2|\Pr[\hat{b} = b] - \frac{1}{2}| = 2\mathsf{Adv}_{\mathsf{SS}}^{\text{2-PK-TTS}}$.

To conclude the proof, due to indistinguishability of PKE-AONT, the adversary distinguishes the ciphertexts in $\mathbf{G}_1$ and the random bit string (as the same length with ciphertexts) with probability $\frac{1}{2} + \mathsf{Adv}_{\mathsf{ind}}^{\text{PKE-AONT}}$.

Hence, by the discussion above and the triangle inequality,

$$\begin{aligned} |\Pr[S_0]| &= |\Pr[S_0] - \Pr[S_1] + \Pr[S_1]| \\ &\leq |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1]| \\ &= 2\mathsf{Adv}_{\mathsf{SS}}^{\text{2-PK-TTS}} + \mathsf{Adv}_{\mathsf{ind}}^{\text{PKE-AONT}} + \frac{1}{2}. \end{aligned}$$

Since $\mathsf{Adv}_{\mathsf{SS}}^{\text{2-TTS}}$ and $\mathsf{Adv}_{\mathsf{ind}}^{\text{PKE-AONT}}$ are two negligible functions of λ, we conclude that the advantage of $\mathcal{A}$ winning the semantic security game is bounded by a negligible function of λ.

Theorem 4. *The scheme Π is traceable against t-collusion under the DDH assumption.*

Proof. By contradiction, assume that there exists an adversary $\mathcal{A}$ that, given the public key BK, t of user keys $\{\mathrm{SK}_{u_1}, \mathrm{SK}_{u_2}, ..., \mathrm{SK}_{u_t}\}$ and an AONT Σ, produces a pirate decoder $\mathcal{D}$ that decrypts all valid ciphertexts perfectly. But when given a probe ciphertext, $\mathcal{D}$ outputs a different value from the pre-computed values in Trace algorithm with non-negligible probabilistic $\epsilon > 0$, i.e., $\Pr[\Sigma(\mathcal{D}(\hat{C}); \rho) \neq M_{k,\sigma}] = \epsilon$. We construct an algorithm $\mathcal{B}$ that breaks the DDH assumption with non-negligible advantage $\frac{\epsilon}{2}$ as follows:

- *Setup.* Algorithm $\mathcal{B}$ is given as input an instance (g, g^u, g^v, X) of DDH assumption, and it wants to determine whether $X = g^{uv}$ or X is a random element in $\mathbb{G}$ ($\mathbb{G}$ has prime order q). $\mathcal{B}$ chooses a position index $k \xleftarrow{\$} \{1, 2, ..., \ell\}$, chooses $f_j(x) = a_{0,j} + a_{1,j}x \pmod{q}$, where $a_{0,j}, a_{1,j} \xleftarrow{\$} \mathbb{Z}_q^*, \forall j \in \{1, 2, ..., \ell\}$, chooses $i_0, i_1, z \xleftarrow{\$} \mathbb{Z}_q^*$ and gives $\mathcal{A}$ $\mathrm{BK} = \langle g, (g^{a_{0,j}}, g^{a_{1,j}})_{j=1}^{\ell} \rangle$ but repalces $g^{a_{0,k}}$ by g^u and $g^{a_{1,k}}$ by $(\frac{g^z}{g^u})^{i_\sigma^{-1}}$, where $\sigma \xleftarrow{\$} \{0, 1\}$. $\mathcal{A}$ chooses a traitor set $\mathsf{T} \subseteq \{1, ..., N\}$ of size t to $\mathcal{B}$. Then $\mathcal{B}$ chooses t codewords $w^{(u_1)}, w^{(u_2)}, ..., w^{(u_t)} \xleftarrow{\$} \Gamma$ (even if Γ is public, the information of which user get which codeword can be hiden, so $\mathcal{B}$ can choose t codewords by his own) satisfy $w_k^{(u_1)} = w_k^{(u_2)} = ... = w_k^{(u_t)} = \sigma$ (the existence of these codewords is guaranteed by the fingerprinting codes) and sets $\mathcal{A}$ the keys

  $$\begin{aligned} \mathrm{SK}_{u_1} &= \langle w^{(u_1)}, i_0, i_1, (f_1(i_{w_1^{(u_1)}}), ..., f_{k-1}(i_{w_{k-1}^{(u_1)}}), z, f_{k+1}(i_{w_{k+1}^{(u_1)}}), ..., f_\ell(i_{w_\ell^{(u_1)}}))\rangle, \\ \mathrm{SK}_{u_2} &= \langle w^{(u_2)}, i_0, i_1, (f_1(i_{w_1^{(u_2)}}), ..., f_{k-1}(i_{w_{k-1}^{(u_2)}}), z, f_{k+1}(i_{w_{k+1}^{(u_2)}}), ..., f_\ell(i_{w_\ell^{(u_2)}}))\rangle, \\ &\vdots \\ \mathrm{SK}_{u_t} &= \langle w^{(u_t)}, i_0, i_1, (f_1(i_{w_1^{(u_t)}}), ..., f_{k-1}(i_{w_{k-1}^{(u_t)}}), z, f_{k+1}(i_{w_{k+1}^{(u_t)}}), ..., f_\ell(i_{w_\ell^{(u_t)}}))\rangle, \end{aligned}$$

  to produces a pirate decoder $\mathcal{D}$.
- *Trace.* By taking a pirate decoder $\mathcal{D}$ as a decryption oracle, $\mathcal{B}$ runs the modified Trace algorithm as follows:

1. Compute $M \xleftarrow{\$} \Sigma(M';\rho)$, where $M' \xleftarrow{\$} \mathcal{M}^{\ell'}$, $\rho \xleftarrow{\$} \{0,1\}^{\tau}$.
2. Choose $j \xleftarrow{\$} \mathbb{Z}_q^*$, where $j \neq i_0$ or i_1. Compute $\bar{c} \leftarrow \langle A_k = m_k X, R = g^v, (j, W_k = X(\frac{(g^v)^z}{X})^{ji_\sigma^{-1}})\rangle$ and set the ciphertext as $\bar{C} \leftarrow \langle k, \bar{c}, Y = \mathsf{MINUS}_k(M)\rangle$.
3. If $\Sigma(\mathcal{D}(\bar{C});\rho) = \mathsf{COMB}_k(Y, A_k/W_k^{\frac{-i_\sigma}{j-i_\sigma}} R^{f_k(i_\sigma)\frac{-j}{i_\sigma - j}})$, $\mathcal{B}$ answers $X = g^{uv}$ or X is a random element in $\mathbb{G}$ randomly; else $\mathcal{B}$ answers X is a random element in $\mathbb{G}$.

If $X = g^{uv}$, ciphertext $\bar{c}$ is a valid ciphertext, since

$$X(\frac{(g^v)^z}{X})^{ji_\sigma^{-1}} = g^{uv}(\frac{(g^v)^z}{g^{uv}})^{ji_\sigma^{-1}} = g^{uv}((\frac{g^z}{g^u})^{i_\sigma^{-1}})^{vj} = g^{uv}(g^{a_{1,k}})^{vj} = g^{v(u+a_{1,k}j)}.$$

In this case, $\Sigma(\mathcal{D}(\bar{C});\rho) = M'_{k,\sigma}$, $\mathcal{B}$ gives the correct answer with probability $\frac{1}{2}$;

If X is a random element in $\mathbb{G}$, the ciphertext $\bar{C}$ is an invalid ciphertext. In this case, $\Sigma(\mathcal{D}(\bar{C});\rho) \neq M'_{k,\sigma}$ with probability ϵ, and $\Sigma(\mathcal{D}(\bar{C});\rho) = M'_{k,\sigma}$ with probability $1-\epsilon$. Therefore $\mathcal{B}$ gives the correct answer with probability $\epsilon + \frac{1}{2}(1-\epsilon) = \frac{1}{2} + \frac{\epsilon}{2}$.

Hence, $\mathcal{B}$ solves the DDH problem with non-negligible advantage $\frac{\epsilon}{2}$. This is a contradiction to the DDH assumption. So we conclude that such adversary $\mathcal{A}$ does not exist.

6 Conclusion

We propose a fully-collusion resistant public-key traitor tracing schemes with efficient black-box tracing. It achieves the asympototically optimal transmission rate for ciphertexts. The storage requirement of each user-key and each public-key are $\ell + 2$ and $2\ell + 1$ respectively, where ℓ is the codeword length of the fingerprinting codes. We show that our scheme is semantically secure based on the DDH hardness assumption and the indistinguishability of the cryptosystem PKE-AONT. Also, our scheme is t-collusion resistant.

There are two open problems. First, How to improve the storage requirements of the user-keys and public-keys further? Second, Billet and Phan [1] proposed a general attack "Pirate 2.0" to attack the code-base traitor tracing schemes. In Pirate 2.0, traitors only give away "part" of user-keys away such that a tracer can trace them with a small probability only. This probability is controlled by traitors according to a specific set of given away keys. How to avoid such attack is also an important problem to make the code-base traitor tracing schemes more practical.

References

1. Billet, O., Phan, D.H.: Traitors collaborating in public: Pirates 2.0. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 189–205. Springer, Heidelberg (2009)

2. Boneh, D., Franklin, M.K.: An efficient public key traitor scheme (Extended abstract). In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999)
3. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM Conference on Computer and Communications Security, pp. 501–510. ACM, New York (2008)
4. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
5. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
6. Boyko, V.: On the security properties of oaep as an all-or-nothing transform. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 503–518. Springer, Heidelberg (1999)
7. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)
8. Chabanne, H., Phan, D.H., Pointcheval, D.: Public traceability in traitor tracing schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 542–558. Springer, Heidelberg (2005)
9. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)
10. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing traitors. IEEE Transactions on Information Theory 46(3), 893–910 (2000)
11. Fazio, N., Nicolosi, A., Phan, D.H.: Traitor tracing with optimal transmission rate. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 71–88. Springer, Heidelberg (2007)
12. Furukawa, J., Attrapadung, N.: Fully collusion resistant black-box traitor revocable broadcast encryption with short private keys. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 496–508. Springer, Heidelberg (2007)
13. Kiayias, A., Yung, M.: Traitor tracing with constant transmission rate. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 450–465. Springer, Heidelberg (2002)
14. Kurosawa, K., Desmedt, Y.G.: Optimum traitor tracing and asymmetric schemes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 145–157. Springer, Heidelberg (1998)
15. Naor, M., Pinkas, B.: Threshold traitor tracing. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 502–517. Springer, Heidelberg (1998)
16. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
17. Phan, D.H.: Traitor tracing for stateful pirate decoders with constant ciphertext rate. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 354–365. Springer, Heidelberg (2006)
18. Phan, D.H., Safavi-Naini, R., Tonien, D.: Generic construction of hybrid public key traitor tracing with full-public-traceability. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 264–275. Springer, Heidelberg (2006)

19. Rivest, R.L.: All-or-nothing encryption and the package transform. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 210–218. Springer, Heidelberg (1997)
20. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC, pp. 116–125. ACM, New York (2003)
21. Tô, V.D., Safavi-Naini, R., Zhang, F.: New traitor tracing schemes using bilinear map. In: Yung, M. (ed.) Digital Rights Management Workshop, pp. 67–76. ACM Press, New York (2003)
22. Tzeng, W.-G., Tzeng, Z.-J.: A public-key traitor tracing scheme with revocation using dynamic shares. Des. Codes Cryptography 35(1), 47–61 (2005)
23. Zhang, R., Hanaoka, G., Imai, H.: On the security of cryptosystems with all-or-nothing transform. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 76–90. Springer, Heidelberg (2004)

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.

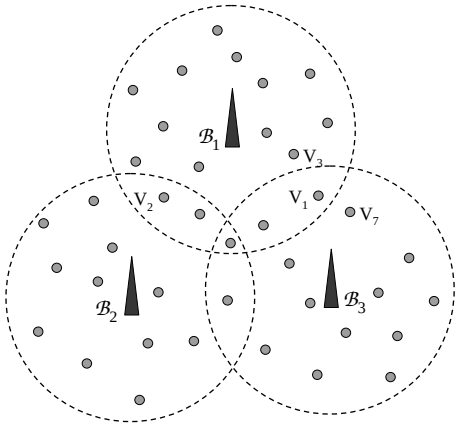

Fig. 1. Deployment of sensor and beacon nodes in a field. Each beacon node uses a different frequency to broadcast random bits and each sensor receives and stores some of them.

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

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

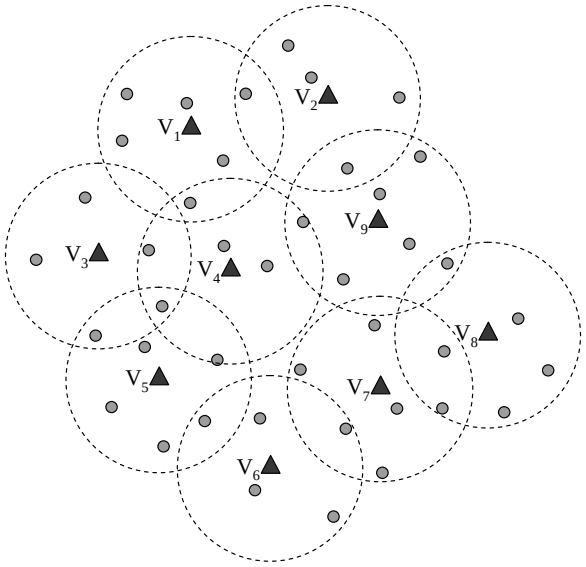

Fig. 3. Deployment of sensor nodes in a field. Some sensor nodes become beaming nodes for broadcasting random bits.

- 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.

Permutation arrays under the Chebyshev distance

Torleiv Kløve, *Fellow, IEEE*, Te-Tsung Lin, Shi-Chun Tsai, *Member, IEEE* and Wen-Guey Tzeng *Member, IEEE*

***Abstract*—An (n, d) permutation array (PA) is a subset of S_n with the property that the distance (under some metric) between any two permutations in the array is at least d. They became popular recently for communication over power lines. Motivated by an application to flash memories, in this paper the metric used is the Chebyshev metric. A number of different constructions are given as well as bounds on the size of such PA.**

***Index Terms*—Permutation arrays, Chebyshev distance, flash memory, code constructions, bounds**

I. Introduction

Let S_n denote the set of all permutations of length n. A permutation array of length n is a subset of S_n. Recently, Jiang et. al [5], [6] showed an interesting new application of permutation arrays for flash memories, where they used different distance metrics to investigate efficient rewriting schemes. Under the multi-level flash memory model, we find the metric induced by the l_∞ norm very appropriate for studying the recharging and error correcting issues. This metric is known as the Chebyshev metric. We consider a noisy channel where pulse amplitude modulation (PAM) is used with different amplitude levels for each permutation symbol. The noise in the channel is an independent Gaussian distribution with zero mean for each position. The received sequence is the original permutation distorted by Gaussian noise, and its ranking can be seen as a permutation, which can be different from the original one.

To study the correlations between ranks, several metrics on permutations were introduced, such as the Hamming distance, the minimum number of transpositions taking one permutation to another, etc. [3], [7]. For instance, Stoll and Kurz [14] investigated a detection scheme of permutation arrays using Spearman's rank correlation. Chadwick and Kurz [2] studied the permutation arrays based on Kendall's tau.

Under the model of additive white Gaussian noise (AWGN) [4], there is a probability for any amplitude level to deviate from the original one, which may yield a large Hamming distance but with a rather small Chebyshev distance. Meanwhile, the original rank may still be in good shape even after some perturbation. Observe that two permutations with a large Hamming distance can actually have a small Chebyshev distance and vice versa. They appear to complement each other in some sense. This inspired us to use the Chebyshev distance. Technically, with l_∞ norm, we find it is much easier to encode, decode and estimate the sphere size of permutation arrays than with the other l_p norms.

In this paper, we give a number of constructions of PAs. For some we give efficient decoding algorithms. We also consider encoding from vectors into permutations.

The research was supported in part by the National Science Council of Taiwan under contracts NSC-95-2221-E-009-094-MY3, NSC-96-2221-E-009-026, NSC-96-2628-E-009-011-MY3, NSC-97-2221-E-009-064-MY3, and NSC-98-2218-E-009-020 and by the Norwegian Research Council. Some of the results of this paper were presented at the 2008 IEEE International Symposium on Information Theory.

T. Kløve is with the Department of Informatics, University of Bergen, N-5020 Bergen, Norway (Email: Torleiv.Klove@ii.uib.no).

T.-T. Lin, S.-C. Tsai and W.-G. Tzeng are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30050, Taiwan (Email: atman.cs94g@nctu.edu.tw, sctsai@csie.nctu.edu.tw, wgtzeng@cs.nctu.edu.tw).

II. Notations

We use $[n]$ to denote the set $\{1, \ldots, n\}$. S_n denotes the set of all permutations of $[n]$. For any set X, X^n denotes the set of all n-tuples with elements from X.

Let id_n denote the identity permutation in S_n. The Chebyshev distance between two permutations $\pi, \sigma \in S_n$ is

$$d_{\max}(\pi, \sigma) = \max\{|\pi_j - \sigma_j| \mid 1 \le j \le n\}.$$

An (n, d) permutation array (PA) is a subset of S_n with the property that the Chebyshev distance between any two distinct permutations in the array is at least d. We sometimes refer to the elements of a PA as code words.

The maximal size of an (n, d) PA is denoted by $P(n, d)$. Let $V(n, d)$ denote the number of permutations in S_n within Chebyshev distance d of the identity permutation. Since $d_{\max}(\mathrm{id}_n, \sigma) = d_{\max}(\pi, \pi\sigma)$, the number of permutations in S_n within Chebyshev distance d of any permutation $\pi \in S_n$ will also be $V(n, d)$. Bounds on $P(n, d)$ and $V(n, d)$ will be considered in Sec. IV.

III. Constructions

In this section we give a number of constructions of PAs, one explicit and some recursive.

A. An explicit construction

Let n and d be given. Define

$$C = \{(\pi_1, \ldots, \pi_n) \in S_n | \pi_i \equiv i \pmod d \text{ for all } i \in [n]\}.$$

Theorem 1: If $n = ad + b$, where $0 \le b < d$, then C is an (n, d) PA and

$$|C| = ((a+1)!)^b (a!)^{d-b}.$$

Proof: Let $1 \le m \le d$ and $u = \lfloor (n-m)/d \rfloor$. For $\pi \in C$, we see that $(\pi_m, \pi_{m+d}, \pi_{m+2d}, \ldots, \pi_{m+ud})$ is a permutation of the set $\{m, m+d, m+2d, \ldots, m+ud\}$. This set contains $(a+1)$ elements if $1 \le m \le b$ and so there are $(a+1)!$ possible choices for $(\pi_m, \pi_{m+d}, \pi_{m+2d}, \ldots, \pi_{m+ud})$ and all can used. Similarly, there are $a!$ choices if $m > b$. Hence the total number of permutations in C is $((a+1)!)^b (a!)^{d-b}$. ■

In particular, we get the following bound.

Theorem 2: If $n = ad + b$, where $0 \le b < d$, then

$$P(n,d) \ge ((a+1)!)^b (a!)^{d-b}.$$

Example 1: For $d = 2$, we get

$$P(2a,2) \ge (a!)^2.$$

We note that if $2d > n$, then $a = 1$ and $b = n - d$ and so $|C| = 2^{n-d}$. If $2d = n$, then $a = 2$, $b = 0$, and we have $|C| = 2^d = 2^{n-d}$ as well. However, if $2d < n$, then $|C| > 2^{n-d}$. Especially, when d is small relative to n, $|C|$ is much larger than 2^{n-d}. For example, for $n = 30, d = 2$, $|C|/2^{n-d} \approx 6.37 \times 10^{15}$.

This construction has a very simple error correcting algorithm. For $d \ge 2t + 1$, we can correct error up to size t in any coordinate. For coordinate i, the codeword has value $\pi_i \equiv i \pmod d$. Suppose that this coordinate is changed into $\sigma = \pi_i + u$, where $|u| \le t$. Then π_i is the integer congruent to i which is closest to σ. Therefore, decoding of position i is done by first computing

$$a \equiv i - \sigma \pmod d,$$

where $-(d-1)/2 \le a \le (d-1)/2$. Then $a = -u$, and so we decode into $\sigma + a = \pi_i$.

B. First recursive construction

Let C be an (n,d) PA of size M, and let $r \ge 2$ be an integer. We define an (rn, rd) PA, C_r, of size M^r as follows: for each multi-set of r code words from C,

$$(\pi_1^{(j)}, \ldots, \pi_n^{(j)}), j = 0, 1, \ldots, r-1,$$

let

$$\rho_j = (r\pi_1^{(j)} - j, \ldots, r\pi_n^{(j)} - j), j = 0, 1, \ldots, r-1,$$

and include $(\rho_0|\rho_1|\ldots|\rho_{r-1})$ as a codeword in C_r. It is clear that under this construction the distance between any two distinct ρ_j, $\rho_{j'}$ is at least rd. It is also easy to check that $(\rho_0|\rho_1|\ldots|\rho_{r-1}) \in S_{rn}$. Hence $|C_r| = M^r$. In particular, we get the following bound.

Theorem 3: If $n > d$ and $r \ge 2$, then

$$P(rn, rd) \ge P(n,d)^r.$$

Proof: Let C be an (n,d) PA of size $P(n,d)$. Then the construction above gives an (rn, rd) PA of C_r. Hence $P(rn, rd) \ge |C_r| = |C|^r = P(n,d)^r$. ∎

C. Second recursive construction

For a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n) \in S_n$ and an integer m, $1 \le m \le n+1$ define

$$\varphi_m(\pi) = (m, \pi_1', \pi_2', \ldots, \pi_n') \in S_{n+1}$$

by

$$\begin{array}{ll} \pi_i' = \pi_i & \text{if } \pi_i \le m, \\ \pi_i' = \pi_i + 1 & \text{if } \pi_i > m. \end{array}$$

Let C be an (n,d) PA, and let

$$1 \le s_1 < s_2 < \cdots < s_t \le n+1$$

be integers. Define

$$C[s_1, s_2, \ldots, s_t] = \{\varphi_{s_j}(\pi) \mid 1 \le j \le t,\ \pi \in C\}.$$

Theorem 4: If C is an (n,d) PA of size M and

$$s_j + d \le s_{j+1} \text{ for } 1 \le j \le t-1,$$

then $C[s_1, s_2, \ldots, s_t]$ is an $(n+1, d)$ PA of size tM.

Theorem 5: If C is an (n,d) PA of size M and $n \le 2d$, then $C[d]$ is an $(n+1, d+1)$ PA of size M.

Proof: If $j > j'$, then

$$d_{\max}(\varphi_{s_j}(\pi), \varphi_{s_{j'}}(\sigma)) \ge s_j - s_{j'} \ge d.$$

Next, consider $j' = j$. If $\pi, \sigma \in C$, $\pi \ne \sigma$, then w.l.o.g, there exist an i such that $\pi_i \ge \sigma_i + d$. Hence

$$d_{\max}(\varphi_{s_j}(\pi), \varphi_{s_j}(\sigma)) \ge \begin{cases} \pi_i - \sigma_i + 1 > d & \text{if } \pi_i > s_j \ge \sigma_i, \\ \pi_i - \sigma_i \ge d & \text{otherwise.} \end{cases}$$

This proves Theorem 4. To complete the proof of Theorem 5 we note that

$$\pi_i \ge \sigma_i + d \ge d + 1 > d,$$

and

$$\sigma_i \le \pi_i - d \le n - d \le d.$$

Hence $\pi_i > d \ge \sigma_i$ and so

$$d_{\max}(\varphi_{s_j}(\pi), \varphi_{s_j}(\sigma)) \ge d + 1.$$

∎

The constructions imply bounds on $P(n,d)$.

Theorem 6: If $n > d \ge 1$, then

$$P(n+1, d) \ge \left(\left\lfloor \frac{n}{d} \right\rfloor + 1\right) P(n,d).$$

Proof: Let $t = \lfloor n/d \rfloor + 1$. Then $(t-1)d + 1 \le n + 1$. If C is an (n,d) PA of size $P(n,d)$, then Theorem 4 implies that $C[1, d+1, 2d+1, \ldots, (t-1)d+1]$ is an $(n+1, d)$ PA of size $tP(n,d)$. Hence $P(n+1,d) \ge tP(n,d)$. ∎

Example 2: In Example 1 we showed that the explicit construction implied that $P(2a,2) \ge (a!)^2$. Combining Theorem 6 and search, we can improve this bound. We have found that $P(7,2) \ge 582$, see the table at the end of the next section. From repeated use of Theorem 6 we get

$$P(2a,2) \ge (a(a-1)\cdots 5)^2 \cdot 4\, P(7,2) \ge \frac{97}{24}(a!)^2.$$

Theorem 5 implies the following bound.

Theorem 7: If $d < n \le 2d$, then

$$P(n+1, d+1) \ge P(n,d).$$

Proof: Let C be an (n,d) PA of size $P(n,d)$. By Theorem 5, $C[d]$ is an $(n+1, d+1)$ PA of size $P(n,d)$. Hence

$$P(n+1, d+1) \ge |C[d]| = P(n,d).$$

∎

Theorem 7 shows in particular that for a fixed r,

$$P(d+1+r, d+1) \ge P(d+r, d) \text{ for } d \ge r. \qquad (1)$$

We will show that $P(d+r, d)$ is bounded. We show the following theorem.

Theorem 8: For fixed r, there exist constants c_r and d_r such that $P(d+r,d) = c_r$ for $d \geq d_r$. Moreover,

$$c_r \leq 2^{2r}\,(2r)! \tag{2}$$

and

$$d_r \leq 1 + (2r-1)c_r - r. \tag{3}$$

Remark. The main point of Theorem 8 is the existence of c_r and d_r. The actual bounds given are probably quite weak in general. For example, Theorem 8 gives the bounds $c_1 \leq 8$ and $d_1 \leq 8$. In Theorem 9 below, we will show that $c_1 = 3$ and $d_1 = 2$. Theorem 8 gives $c_2 \leq 384$ and $d_2 \leq 1151$, whereas numerical computation indicate that $c_2 = 9$ and $d_2 = 5$.

We split the proof of Theorem 8 into three lemma.

Lemma 1: If $d \geq r$, then $P(d+r,d) \leq 2^{2r}\,(2r)!$.

Proof: Suppose that there exists an $(d+r,d)$ PA C of size $M > 2^{2r}\,(2r)!$. We call the integers

$$1, 2, \ldots, r \text{ and } d+1, d+2, \ldots, d+r$$

potent, the first r *smaller potent*, the last r *larger potent*. Two potent integers are called *equipotent* if both are smaller potent or both are larger potent. If the distance between two permutations $(\pi_1, \pi_2, \ldots, \pi_n)$, $(\rho_1, \rho_2, \ldots, \rho_n)$ is at least d, then there exists some position i such that, w.l.o.g, $\pi_i - \rho_i \geq d$, Then π_i is a larger potent element and ρ_i is smaller potent. Each permutation in S_{d+r} contains $2r$ potent elements and we call the set of positions of these the *potency support* $\chi(\pi)$ of the permutation, that is, the potency support of π is

$$\chi(\pi) = \{i \mid 1 \leq \pi_i \leq r\} \cup \{i \mid d+1 \leq \pi_i \leq d+r\}.$$

The potency support of C is the union of the potency support of the permutations in C, that is

$$\begin{aligned}\chi(C) =& \{i \mid 1 \leq \pi_i \leq r \text{ for some } \pi \in C\} \\ &\cup \{i \mid d+1 \leq \pi_i \leq d+r \text{ for some } \pi \in C\}.\end{aligned}$$

Let $\pi \in C$. For each $\rho \in C$, $\rho \neq \pi$, we have $d(\pi,\rho) \geq d$. Hence there exists some $i \in \chi(\pi)$ such that ρ_i is potent. Therefore, the set

$$\{(\rho, i) \mid \rho \in C \text{ and } i \in \chi(\pi)\}$$

contains at least $2r + (M-1) > M$ elements. Hence there is an $i \in \chi(\pi)$ such that

$$|\{\rho \in C \mid \rho_i \text{ is potent}\}| > M/(2r) > 2^{2r}(2r-1)!.$$

Since

$$\begin{aligned}\{\rho \in C \mid \rho_i \text{ is potent}\} =& \{\rho \in C \mid \rho_i \text{ is smaller potent}\} \\ &\cup \{\rho \in C \mid \rho_i \text{ is larger potent}\},\end{aligned}$$

there exists a subset $C_1 \subset C$ such that

$$|C_1| > 2^{2r-1}(2r-1)!$$

and the elements in position $i_1 = i$ are equipotent.

We can now repeat the procedure. Let $\pi \in C_1$. There must exist an $i_2 \in \chi(\pi) \setminus \{i_1\}$ such that

$$|\{\rho \in C_1 \mid \rho_{i_2} \text{ is potent}\}| \geq |C_1|/(2r-1) > 2^{2r-1}(2r-2)!.$$

Hence we get subset $C_2 \subset C_1$ such that

$$|C_2| > 2^{2r-2}(2r-2)!$$

and the elements in position i_2 are equipotent (and the elements in position i_1 are equipotent).

Repeated use of the same argument will produce for each j, $1 \leq j \leq 2r$ a set C_j such that

$$|C_j| > 2^{2r-j}\,(2r-j)!$$

and for j positions $i_1, i_2, \ldots i_j$, the elements in those positions are all equipotent. In particular, $|C_{2r}| > 1$, all permutations in C_{2r} have the same potency support $\{i_1, i_2, \ldots, i_{2r}\}$, and for each of these positions, all the elements in that position are equipotent. This is a contradiction since the distance between two such permutations must be less than d. Hence the assumption that a PA of size larger than $2^{2r}\,(2r)!$ exists leads to a contradiction. ∎

Lemma 1 combined with (1) proves the existence of c_r and d_r and gives the bound (2).

Lemma 2: If C is a $(d+r,d)$ PA of size M where

$$d > r \text{ and } d + r > |\chi(C)|,$$

then there exists a $(d-1+r, d-1)$ PA of size M. In particular, if $M = P(d+r,d)$, then

$$P(d-1+r, d-1) = P(d+r,d).$$

Proof: Replace all elements in range $r+1, r+2, \ldots, d$ in the permutations of C by a star $*$ which will denote "unspecified". The permutations in C is transformed into *vectors* containing the potent elements and $d-r$ stars. Note that if we replace the unspecified elements in each vector by the integers $r+1, r+2, \ldots, d$ in some order, we get a permutation, and the distance between two such permutations will be at least d since we have not changed the potent elements.

Since the length $d+r$ of C is larger than $|\chi(C)|$, there exists a position where all the vectors contains a star. Remove this position from each vector and reduce all the larger potent elements by one. This given a set of M vectors of length $d-1+r$ and such that the distance between any two is at least $d-1$. Replacing the $d-1-r$ stars in each vector by $r+1, r+1, \ldots, d-1$ in some order, we get a $(d-1+r, d-1)$ PA of size M.

If $M = P(d+r,d)$, then we get

$$P(d-1+r, d-1) \geq P(d+r,d).$$

Since $P(d-1+r, d-1) \leq P(d+r,d)$ by (1), the lemma follows. ∎

Lemma 3: If C is a $(d+r,d)$ PA of size M and $d \geq r$, then

$$|\chi(C)| \leq M(2r-1) + 1.$$

Proof: Each permutation has potency support of size $2r$. The potency support of any two permutations in C must overlap since their distance is at least d. Hence each permutation after the first will contribute at most $2r-1$ new elements to the total potency support. Therefore,

$$|\chi(C)| \leq 2r + (M-1)(2r-1).$$

■

Remark. By a more involved analysis, we can improve this bound somewhat. For example, we see that two new permutations can contribute at most $4r-3$ to the total support.

We can now complete the proof of Theorem 8. Let C be a $(d+r, r)$ code of size c_r. By Lemma 3, $|\chi(C)| \leq c_r(2r-1)+1$. If $d > 1 + c_r(2r-1) - r$, then $d + r > |\chi(C)|$. Hence, by Lemma 2, $P(d-1+r, d-1) = P(d+r, d)$. Therefore, $d_r \leq 1+c_r(2r-1)-r$, that is, (3) is satisfied. This completes the proof of Theorem 8.

Theorem 9: We have $P(d+1, d) = 3$ for $d \geq 2$.

Proof: We use the same notation as in the proof of Lemma 2. Let C be an $(d+1, d)$ PA. The only potent elements are 1 and n. W.lo.g. we may assume the first permutation in C is $(1, n, *, *, \ldots)$ where $*$ denotes some unspecified integer in the range $2, 3, \ldots, d$. W.l.o.g, a second permutation has one of three forms:

$$(n, 1, *, *, \ldots),\ (n, *, 1, *, \ldots),\ (*, 1, n, *, \ldots).$$

We see that if the second permutation is of the first form, there cannot be more permutations. If the second permutation is of the form $(n, *, 1, *, \ldots)$, then there is only one possible form for a third permutation, namely $(1, *, n, *, \ldots)$. Hence we see that $P(d+1, d) \leq 3$ and that $P(d+1, d) = 3$ for $d \geq 2$. ■

To determine $P(d+r, d)$ along the same lines for $r \geq 2$ seems to be difficult because of the many cases that have to be considered. Even to determine $P(d+2, d)$ will involve a large number of cases. For example for the second permutation there are 138 essentially different possibilities for the four positions in the potency support of the first permutation. For each of these there are many possible third permutations, etc.

D. Encoding/decoding of some PA constructed by the second recursive construction

Suppose we start with the PA

$$C_d = \{(1, 2, 3, \ldots, d)\}.$$

For $\nu = d, d+1, \ldots, n-1$ let

$$C_{\nu+1} = C_\nu[1, \nu+1].$$

Then C_n is an (n, d) PA of size 2^{n-d}. For some applications, we may want to map a set of binary vectors to a permutation array. One algorithm for mapping a binary vector $(x_1, x_2, \ldots, x_{n-d})$ into C_n would be to use the recursive construction of C_n by mapping $(x_1, x_2, \ldots, x_i)$ into a permutation π in C_{d+i}. Recursively, we can then map $(x_1, x_2, \ldots, x_i, 0)$ to $\varphi_1(\pi)$ and $(x_1, x_2, \ldots, x_i, 1)$ to $\varphi_{d+i+1}(\pi)$.

However, there is an alternative algorithm which requires less work. Retracing the steps of the construction, we see that given some initial part of length less than $n-d$ of a permutation in C_n, there are exactly two possibilities for the next element, one "larger" and one "smaller". More precisely, induction shows that if the initial part of length $i-1$ contains exactly t "smaller" elements, then element number i is either $t+1$ (the "smaller") or $n-i+t+1$ (the "larger"). This is the basis for a simple mapping from Z_2^{n-d} to C_n. We give this algorithm in Figure 1.

Input: $(x_1, \ldots, x_{n-d}) \in Z_2^{n-d}$
Output: $(\pi_1, \ldots, \pi_n) \in C_n$
  for $i \leftarrow n-d+1$ **to** n **do** $x_i \leftarrow 0$;
  $t \leftarrow 0$; //* t is the number of zeros seen so far.*//
  for $i \leftarrow 1$ **to** n **do**
    if $x_i = 0$
      then $\{\pi_i \leftarrow t+1;\ t \leftarrow t+1;\}$
      else $\{\pi_i \leftarrow n-i+t+1;\}$

Fig. 1. Algorithm mapping Z_2^{n-d} to C_n

We see that the difference between the larger and the smaller element in position $i \leq n-d$ is $n-i$. Hence we can recover from any error of size less than $(n-i)/2$ by choosing the closest of the two possible values, and the corresponding binary value. We give the decoding algorithm in Figure 2.

Input: $(\pi_1, \ldots, \pi_n) \in [n]^n$
Output: $(x_1, \ldots, x_{n-d})$
  $t \leftarrow 0$; //* t is number of zeros determined. *//
  for $i \leftarrow 1$ **to** $n-d$ **do**
    if $\pi_i < (n-i)/2 + t + 1$
      then $\{x_i \leftarrow 0;\ t \leftarrow t+1;\}$
      else $\{x_i \leftarrow 1;\}$

Fig. 2. Decoding algorithm recovering the binary preimage from a corrupted permutation in C_n.

Without going into all details, we see that we can get a similar mapping from q-ary vectors. Now we start with the PA

$$C_{(q-1)d} = \{(1, 2, 3, \ldots, (q-1)d)\}.$$

For $(q-1)d \leq \nu \leq n-1$ let $s_j = (j-1)\lfloor \nu/(q-1) \rfloor + 1$ for $1 \leq j \leq q-1$ and $s_q = \nu + 1$. Let

$$C_{\nu+1} = C_\nu[s_1, s_2, \ldots, s_q].$$

Then C_n is an (n, d) PA of size $q^{n-(q-1)d}$. Encoding and decoding correcting errors of size at most $(d-1)/2$, based on the recursion, is again relatively simple.

IV. Further bounds on $P(n, d)$

A. General bounds

Since $d_{\max}(\pi, \sigma) \leq n-1$ for any two distinct permutations in S_n, we have $P(n, n) = 1$. Therefore, we only consider $d < n$.

Since the spheres of radius d in S_n all have size $V(n, d)$, we can get a Gilbert type lower bound on $P(n, d)$.

Theorem 10: For $n > d \geq 2$ we have

$$P(n, d) \geq \frac{n!}{V(n, d-1)}.$$

Proof: It is clear that the following greedy algorithm produces a permutation array with cardinality at least $n!/V(n, d-1)$.

1) Start with any permutation in S_n.
2) Choose a permutation whose distance is at least d to all previous chosen permutations.
3) Repeat step 2 as long as such a permutation exists.

Let C be the permutation array produced by the above greedy algorithm. Once the algorithm stops, S_n will be covered by the $|C|$ spheres of radius $d-1$ centered at the code words in C. Thus $n! \le |P| \cdot V(n, d-1)$ which implies our claim. ∎

Similarly, we get a Hamming type upper bound in the usual way.

Theorem 11: If $n > d \ge 1$, then

$$P(n,d) \le \frac{n!}{V(n, \lfloor (d-1)/2 \rfloor)}.$$

Proof: Let C be an (n,d) PA of size $P(n,d)$. The spheres of radius $\lfloor (d-1)/2 \rfloor$ around the permutations in C are pairwise disjoint. The union of these spheres is a subset of S_n. Hence

$$P(n,d)V(n, \lfloor (d-1)/2 \rfloor) = |C|V(n, \lfloor (d-1)/2 \rfloor) \le n!$$

and the bound follows. ∎

If $n \le 2d$ and d is even, we can combine the bound in Theorem 11 with Theorem 7 to get the following bound which is stronger than the ordinary Hamming bound, at least in the cases we have tested.

Theorem 12: If d is even and $2d \ge n > d \ge 2$, then

$$P(n,d) \le \frac{(n+1)!}{V(n+1, d/2)}.$$

Proof:

$$P(n,d) \le P(n+1, d+1) \le \frac{(n+1)!}{V(n, d/2)}.$$

∎

Example 3: For $n = 11$ and $d = 6$, Theorem 11 gives

$$P(11,6) \le \left\lfloor \frac{11!}{V(11,2)} \right\rfloor = \left\lfloor \frac{11!}{11854} \right\rfloor = 3367$$

whereas Theorem 12 gives

$$P(11,6) \le \left\lfloor \frac{12!}{V(12,3)} \right\rfloor = \left\lfloor \frac{12!}{563172} \right\rfloor = 850.$$

Remark. We can of course use Theorem 7 repeatedly r times and then Theorem 11 to get

$$P(n,d) \le \frac{(n+r)!}{V(n+r, \lfloor (d+r-1)/2 \rfloor)}$$

for all $r \ge 0$. However, it appears we get the best bounds for $r = 1$ when d is even and $r = 0$ when d is odd.

In general, no simple expression of $V(n,d)$ is known. A survey of known results as well as a number of new results on $V(n,d)$ were given by Kløve [8]. See also Kløve [9] and [10]. Here we briefly give some main results.

As observed by Lehmer [11], $V(n,d)$ can be expressed as a permanent. The permanent of an $n \times n$ matrix A is defined by

$$\mathrm{per}A = \sum_{\pi \in S_n} a_{1,\pi_1} \cdots a_{n,\pi_n}.$$

In particular, if A is a $(0,1)$-matrix, then

$$\mathrm{per}A = |\{\pi \in S_n : a_{i,\pi_i} = 1 \text{ for all } i\}|.$$

Let $A^{(n,d)}$ be the $n \times n$ matrix with $a_{i,j}^{(n,d)} = 1$ if $|i-j| \le d$ and $a_{i,j}^{(n,d)} = 0$ otherwise.

Lemma 4: $V(n,d) = \mathrm{per}A^{(n,d)}$.

Proof:

$$\begin{aligned} V(n,d) &= |\{\pi \in S_n : d_{\max}(\mathrm{id}, \pi) \le d\}| \\ &= |\{\pi \in S_n : |\pi_i - i| \le d \text{ for all } i\}| \\ &= |\{\pi \in S_n : a_{i,\pi_i}^{(n,d)} = 1 \text{ for all } i\}| \\ &= \mathrm{per}A^{(n,d)}. \end{aligned}$$

∎

For fixed d, $V(n,d)$ satisfies a linear recurrence in n. A proof is given in [13] (Proposition 4.7.8 on page 246). For $1 \le d \le 3$ these recurrences were determined explicitly by Lehmer [11], and for $4 \le d \le 6$ by Kløve [8]. In particular, this implies that

$$\lim_{n \to \infty} V(n,d)^{1/n} = \mu_d,$$

where μ_d is the largest root of the minimal polynomial corresponding to the linear recurrence of $V(n,d)$. Lehmer [11] determined μ_d approximately for $d = 1, 2, 3$ and Kløve [8] for $d \le 8$.

For an $n \times n$ $(0,1)$-matrix it is known (see Theorem 11.5 in [16]) that

$$\mathrm{per}A \le \prod_{i=1}^{n} (r_i!)^{1/r_i},$$

where r_i is the number of ones in row i.

For $A^{(n,d)}$ we clearly have $r_i \le 2d+1$ for all i. Hence

$$V(n,d) \le [(2d+1)!]^{n/(2d+1)} \text{ for all } n \tag{4}$$

and

$$\mu_d \le [(2d+1)!]^{1/(2d+1)}.$$

In Table I we give μ_d and this upper bound.

TABLE I
μ_d AND ITS UPPER BOUND.

d	μ_d	$[(2d+1)!]^{1/(2d+1)}$	$\mu_d/(2d+1)$
1	1.61803	1.81712	0.53934
2	2.33355	2.60517	0.46671
3	3.06177	3.38002	0.43739
4	3.79352	4.14717	0.42150
5	4.52677	4.90924	0.41152
6	5.26082	5.66769	0.40468
7	5.99534	6.42342	0.39969
8	6.73016	7.17704	0.39589

We note that for large d, $\mu_d/(2d+1) \approx 1/e \approx 0.36788$.

Combining Theorem 10 and (4) we get

Corollary 1: For $n > d \ge 1$, we have

$$P(n,d) \ge \frac{n!}{[(2d-1)!]^{n/(2d-1)}}.$$

Combining equations (33) and (34) in Kløve [8] we get the following lower bound on $V(n,d)$:

$$V(n,d) \geq \frac{n!\,(2d+1)^n}{2^{2d}\,n^n}. \tag{5}$$

For d odd, (5) gives

$$V(n,\lfloor (d-1)/2 \rfloor) = V(n,(d-1)/2) \geq \frac{n!\,d^n}{2^{d-1}\,n^n}.$$

Combining this with Theorem 11 we get the following explicit upper bound on $P(n,d)$.

Corollary 2: For d odd and $n > d \geq 1$, we have

$$P(n,d) \leq \frac{2^{d-1}n^n}{d^n}.$$

Similarly, for d even, combining Theorem 11 and Theorem 12 with (5), we get the following.

Corollary 3: For d even and $n > d \geq 2$, we have

$$P(n,d) \leq \min\left\{ \frac{2^{d-2}n^n}{(d-1)^n}, \frac{2^d(n+1)^{n+1}}{(d+1)^{n+1}} \right\}.$$

The bounds on $V(n,d)$, both the upper and the lower, are in most cases quite weak and so the bounds on $P(n,d)$ also become quite weak.

B. Table of bounds on $P(n,d)$

We have used the following greedy algorithm to find an (n,d) PA C: Let the identity permutation in S_n be the first permutation in C. For any set of permutations chosen, choose as the next permutation in C the lexicographically next permutation in S_n with distance at least d to the chosen permutations in C if such a permutation exists. The size of the resulting PA is of course a lower bound on $P(n,d)$.

The lower bounds in Table II were in most cases found by this greedy algorithm. For $n = 8$, $d = 5$, the greedy algorithm gave a PA of size 26. However,

$$P(8,5) \geq P(7,4) \geq 28$$

by Theorem 7. Similarly,

$$P(10,7) \geq P(9,6) \geq P(8,5) \geq 28.$$

Some other of the lower bounds are also determined using Theorem 7. They are marked by $*$. The upper bound is the Hamming type bound in Theorem 11 or it's modified bound in Theorem 12. Since $P(n,1) = n!$ for all n, this is not included in the table.

TABLE II
BOUNDS ON $P(n,d)$.

	$d=2$	$d=3$	$d=4$
$n=d+1$	3	3	3
$n=d+2$	$6-24$	9	$9-12$
$n=d+3$	$29-120$	$20-34$	$28-43$
$n=d+4$	$90-720$	$84-148$	$68-166$
$n=d+5$	$582-5040$	$401-733$	$283-4077$

	$d=5$	$d=6$	$d=7$
$n=d+1$	3	3	3
$n=d+2$	$9-12$	$9-18$	$9-18$
$n=d+3$	28^*-43	28^*-60	28^*-60
$n=d+4$	$95-166$	95^*-216	95^*-216
$n=d+5$	$236-714$	236^*-850	236^*-850

V. CONCLUSION

We give a number of constructions of permutations arrays under the Chebyshev distance, some with efficient error correction algorithms. We also consider an explicit mapping of vectors to permutations with efficient encoding/decoding. Finally, we give some bounds on the size of PAs under the Chebyshev distance.

Tamo and Schwartz [15] independently considered this problem and gave, among other results, a construction equivalent to our first construction as well as some other constructions.

REFERENCES

[1] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash memories*. Kluwer Academic Publishers, 1999.

[2] H. Chadwick, L. Kurz, "Rank permutation group codes based on Kendall's correlation statistic," *IEEE Trans. Inform. Th.,* vol. IT-15, pp. 306–315, Mar 1969.

[3] P. Diaconis, *Group Representations in Probability and Statistics*. Hayward, CA: Institute of Mathematical Statistics, 1988.

[4] S. Haykin, *Communication Systems*, 4th Ed. John Wiley & Sons, 2001.

[5] A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank Modulation for Flash Memories," in *Proc. IEEE Internat. Symp. on Inform. Th.*, 2008, pp. 1731-1735.

[6] A. Jiang, M. Schwartz and J. Bruck, "Error-Correcting Codes for Rank Modulation," in *Proc. IEEE Internat. Symp. on Inform. Th.*, 2008, pp. 1736-1740.

[7] M. Kendall and J. D. Gibbons, *Rank correlation methods.* London, U.K.: Edward Arnold, 1990.

[8] T. Kløve, "Spheres of Permutations under the infinity norm - permutations with limited displacement," Reports in Informatics, Dept. of Informatics, Univ. Bergen, Report no. 376, November 2008.

[9] T. Kløve, "Generating functions for the number of permutations with limited displacement," The Electronic Journal of Combinatorics, R104, vol. 16(1), August 14, 2009.

[10] T. Kløve, "Lower bounds on the size of spheres of permutations under the Chebychev distance," *Designs, Codes and Cryptography*, to appear.

[11] D. H. Lehmer, "Permutations with strongly restricted displacements," in *Combinatorial Theory and its Applications II*, P. Erdös, A. Rńyi and V. T. Sós (eds.), Amsterdam: North Holland Publ., 1970.

[12] K. W. Shum, "Permutation coding and MFSK modulation for frequency selective channel," *IEEE Personal, Indoor and Mobile Radio Communications*, vol. 13, pp. 2063–2066, Sept. 2002.

[13] R. P. Stanley, *Enumerative Combinatorics*, Vol. I. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[14] E. Stoll and L. Kurz, "Suboptimum Rank Detection Procedures Using Rank Vector Codes," *IEEE Trans. Commun.*, vol. COM-16, pp. 402-410, June 1968.

[15] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," arXiv:0907.3387, July 20, 2009.

[16] J.H. van Lint, R. M. Wilson, *A Course in Combinatorics*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2001.

Torleiv Kløve (M'89–SM'91–F'03) was born in Voss, Norway, 1943. He received the Cand. Mag., Cand. Real., and Dr. Philos. degrees from the University of Bergen, Norway, in 1966, 1967, and 1971, respectively.

He has been with the University of Bergen since 1971, first as Senior Lecturer in Mathematics, and, since 1982, Professor of Informatics. During the academic years 1975–1976, 1981–1982, and 1990–1991 he spent sabbaticals at the University of Hawaii at Manoa. During the academic year 2001–2002, he was a Visiting Professor at The Chinese University of Hong Kong, and during the fall semester 2002, he was Visiting Professor at Hong Kong University of Science and Technology. During the academic year 2008–2009 he spent a sabbatical at the University of California, Santa Cruz.

His research interests include coding theory, number theory, and combinatorics. He is coauthor of the book Error Detecting Codes (Kluwer 1995) and the author of the book Codes for Error Detection (World Scientific 2007).

Prof. Kløve was Chairman of the 1996 IEEE Information Theory Workshop, Longyearbyen, Norway, Associate Editor for Coding Theory for IEEE TRANSACTIONS ON INFORMATION THEORY (1996–1999), and member of the Board of Governors of the IEEE Information Theory Society (2001–2003).

Te-Tsung Lin received his BS and MS degrees in Computer Science from National Chiao Tung University, Taiwan, in 2005 and 2007, respectively. His research interests include Combinatorics, Coding theory and Algorithms.

Shi-Chun Tsai received his BS and MS degrees in Computer Science and Information Engineering from National Taiwan University, Taiwan, in 1984 and 1988, respectively; and Ph.D. degree in Computer Science from the University of Chicago, USA, in 1996. During 1993-1996, he served as Lecturer in Computer Science Department, the University of Chicago. During 1996-2001, he was Associate Professor of Information Management Department and Computer Science and Information Engineering Department, National Chi Nan University, Taiwan. He has been with the Department of Computer Science, National Chiao Tung University, Taiwan since 2001, and was promoted to full Professor in 2007. His research interests include Computational Complexity, Algorithms, Coding theory and Combinatorics.

Wen-Guey Tzeng received his BS degree in Computer Science and Information Engineering from National Taiwan University, Taiwan, 1985; and MS and Ph.D. degrees in Computer Science from the State University of New York at Stony Brook, USA, in 1987 and 1991, respectively. He joined the Department of Computer and Information Science (now, Department of Computer Science), National Chiao Tung University, Taiwan, in 1991. Dr Tzeng now serves as Chairman of the department. His current research interests include Cryptology, Information Security and Network Security.