行政院國家科學委員會專題研究計畫 成果報告

基於生體特徵與暗號的免憑證密碼系統的研究與開發
研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 97-2218-E-004-002-
執行期間：97年08月01日至98年07月31日
執行單位：國立政治大學資訊科學系

計畫主持人：左瑞麟

計畫參與人員：碩士班研究生-兼任助理人員：詹省三
碩士班研究生-兼任助理人員：陳力瑋
碩士班研究生-兼任助理人員：陳淵順

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 98年10月28日

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

基於生物認證與暗號的免憑證密碼體系的研究與開發

計畫類別：■個別型計畫 □ 整合型計畫
計畫編號：NSC － 97 － 2218 － E － 004 － 002
執行期間： 97 年 8 月 1日至 98 年 7 月 31 日

計畫主持人：左瑞麟
共同主持人：
計畫參與人員： 陳力瑋（兼任助理)、陳淵順（兼任助理)、
詹省三（兼任助理）

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：國立政治大學

中 華 民 國 八十九 年 十 月 二十五 日

中文摘要：

計畫名稱:基於生物認證與暗號的免憑證密碼體系的研究與開發

關鍵詞:資訊安全，密碼學，生物認證，免憑證密碼系統，通行碼

中文摘要
本計畫的目的在研究與開發安全性高而且可以兼顧使用便利性的密碼系統。此密碼系統利用生體特徵(指紋)為秘密金鑰，欲加密文件或做電子簽名時，只需要將指紋輸入本計畫開發的系統中即可。利用方法和現實社會中按壓指紋或蓋印章非常相近，因此，不需複雜的操作及艱深的密碼學知識，任何人皆可輕鬆上手，完成文件加密或電子簽名。國內外對於生物特徵利用在密碼學的現行研究方面，目前仍停留在個人的身分認證上面，利用生物特徵來做明文加解密或電子簽名的可謂少之又少。其原因在於指紋等的生物特徵屬於個人隱私，而保護此個人隱私在現行的研究上有其難度(因為在現行研究中，生物特徵若做為使用者的密鑰，必須事先登錄在PKI(公開金鑰基礎結構)或系統管理者伺服器內，因此有個人隱私洩漏的問題)。本計畫利用免憑證密碼系統，除了解決了個人隱私可能洩漏的問題之外，因為認證中心不產生憑證，所以在簡化PKI上也是值得期待的。本計畫的實現在保護個人隱私及資訊安全方面，為使用者提供了一個更安全且方便的方式。

在技術成果方面，本計畫實際做出了需同時利用指紋及密鑰才能做出數位簽名或解密的系統。此系統實際上利用免憑證密碼系統為理論基礎。在免憑證密碼系統的理論研究部分，本計畫最終提出了三項理論研究成果並發表於國內外學術會議中。

Abstract:

Title: A Research and Development on Biometric and Password-Based Certificateless Cryptosystem

Key words: Biometrics, Certificateless cryptosystem, Cryptography, Information Security, Password

Abstract

The purpose of the project is to design and produce a secure and high-performance cryptographic device. The system of the device utilizes a user's bio-information (ie., the fingerprint) as his private key. In this way, the user does not have to remember a long and high entropy private key. In addition, to sign a document or to encrypt a message can be done very easily by just scan his fingerprint into our system (using the device we produced). This is very similar to take a person's fingerprint in real world and the user does not have to have the knowledge about cryptography. We notice that current researches about biometrics enable us to use finger to help authenticate us to a computer or network but not to help sign or encrypt a document. The difficulty of signing or encrypting a document using biometrics is the problem of personal information leakage (ie., user's bio-information has to be registered in PKI or stored at servers previously). We solve this problem by using a certificateless cryptosystem. Our project can not only produce a secure and high-performance cryptosystem on one hand but can also protect personal information on the other hand.

Using a fingerprint reader as an assistant device, this project finally designed such a cryptographic device that can use both a fingerprint and a secret key to sign or to decrypt a message. On the other hand, the theoretic research is focused on certificateless cryptosystems. We have one result that has been published and presented at an international conference and have two results that have been accepted by a domestic symposium.

報告內容：

前言

隨著電腦與網路科技的進步，許多的資訊可以透過網際網路瞬間取得或與他人共享。如此一來，無形之中衍生了許多竊密、詐騙或偽造資訊等等的網路犯罪行為。因此，資訊安全與保護個人隱私的技術已成為時下最重要的課題。密碼學保障了資訊在網路上的傳輸不至被竊聽或盜取內容，保障了資訊的隱密性，也保障了資訊的合法性。因此，密碼可以說是為廣大的計算機及網路的使用者提供安全服務的一項重要技術。密碼學的歷史雖然很悠久，可是，過去一直是利用在軍事上的用途。一直到1980年代左右，才真正普及到商業上的用途或個人的使用者。因此，對於電腦技術不甚瞭解的使用者，尤其是上了年紀的使用者或電腦初學者來說，密碼仍然是個難學又難用的技術。另一方面，對於密碼系統的管理者而言，以公開金鑰基礎結構(PKI)為使用前提的公開金鑰密碼體系，存在有憑證註銷的問題。為了解決此問題，所以額外負擔了很高的管理成本。針對這樣的情況，本計畫希望在學術及實際應用兩方面做出貢獻，提出有效的解決方案。

研究目的

本計畫的目的在研究與開發安全性高而且可以兼顧使用便利性的密碼系統。此密碼系統利用生體特徵(指紋)為秘密金鑰，欲加密文件或做電子簽名時，只需要將指紋輸入本計畫開發的系統中即可。利用方法和現實社會中按壓指紋或蓋印章非常相近，因此，不需複雜的操作及艱深的密碼學知識，任何人皆可輕鬆上手，完成文件加密或電子簽名。國內外對於生物特徵利用在密碼學的現行研究方面，目前仍停留在個人的身分認證上面，利用生物特徵來做明文加解密或電子簽名的可謂少之又少。其原因在於指紋等的生物特徵屬於個人隱私，而保護此個人隱私在現行的研究上有其難度(因為在現行研究中，生物特徵若做為使用者的密鑰，必須事先登錄在PKI(公開金鑰基礎結構)或系統管理者伺服器內，因此有個人隱私洩漏的問題)。本計畫利用免憑證密碼系統，除了解決了個人隱私可能洩漏的問題之外，因為認證中心不產生憑證，所以在簡化PKI上也是值得期待的。本計畫的實現在保護個人隱私及資訊安全方面，為使用者提供了一個更安全且方便的方式。另外，本計畫的密碼理論部分，利用免憑證密碼系統為基礎。免憑證密碼系統的理論研究將為本計畫的另一項研究重點。

文獻探討

首先，針對公開金鑰密碼體系作個簡單的介紹。公開金鑰密碼系統(Public key Cryotosystem)表示系統（或使用者）的加密與解密中使用的鑰匙是不同的。加密用的鑰匙稱為公開金鑰(Public Key)，他可以在網站或公開目錄上公布給他人知道，甚至是愈多人知道愈好;另一把稱為秘密金鑰(Private Key)，持有人必須完全隱藏著這把鑰匙，絕對不可洩漏給他人知道。因為加密用的金鑰可以公開，而且和系統的使用者人數無關，每個人只需一把公開金鑰及相對應的秘密金鑰即可，因此和傳統的密碼系統（加解密用相同金鑰的對稱式密碼系統）相比較，金鑰的管理相對容易許多，安全性也更高。另外，公開金鑰密碼系統的另一項特徵是可以應用在電子簽名上，以達到確認送信者、防止偽造及防止竄改的功能。但是，公開金鑰密碼系統的信賴性是建立在公開金鑰基礎結構之上。公開金

鑰基礎結構(Public Key Infrastructure, PKI)是由管理電子憑證的許多機構所組成。其中最重要的組成元素即為驗證中心(Certificate Authority, CA)。CA的主要工作就是簽發金鑰憑證，以提供系統使用者能取得所需他人的認證資料。此外，CA亦需維護憑證資料庫以及定期發佈憑證註銷清單(Certification Revocation List)。PKI在實作及運用上衍生出了序多問題，如金鑰憑證數量易於過度激增及憑證註銷清單過大等問題。這些問題不只降低PKI運作的效率，也增加了管理的成本。

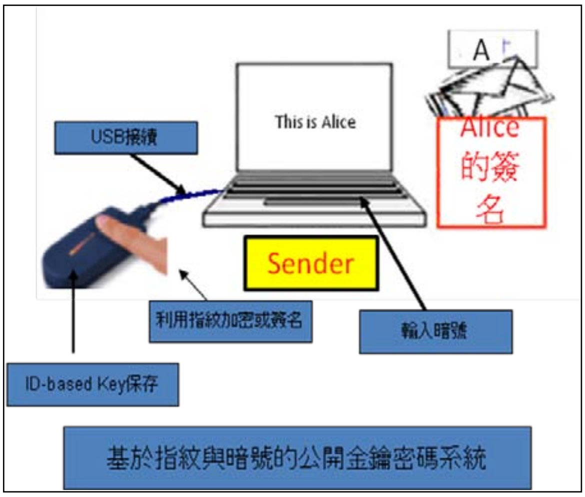
1984年Shamir[5]開創性的提出了基於身份(Identity based, ID-based)的公開金鑰密碼體系。在該體系中，Shamir建議使用能標示使用者身份的信息作為公開金鑰，譬如使用者的姓名或E-mail地址。基於身份的密碼體系排除了對金鑰憑證的需要和依賴，在一定程度上解決了現行PKI所遇到的問題。但是，在基於身份的公開金鑰密碼體系當中，使用者的秘密金鑰不能由使用者自己產生，而必須完全依賴密鑰生成中心(Private Key Generator,PKG)來產生。因此PKG必須完全值得信賴，而有了PKG信賴性問題。

為了完全解決利用PKI所產生的金鑰憑證數量過度激增及憑證註銷清單過大等問題以及利用基於身份的密碼體系所產生的PKG信賴問題，Al-Riyami和Paterson[1]於2003年提出了免憑證公開金鑰密碼系統(Certificateless Public Key Crptosystem)。免憑證公開金鑰密碼系統同時繼承了傳統公開金鑰密碼系統及基於身份的密碼系統的優點。在金鑰生成方面，公開金鑰以及對應的秘密金鑰可由使用者自己產生。公開金鑰雖然仍需要信賴機關CA的認證，但CA並不簽發金鑰憑證，而是利用基於身份的密碼系統的方式產生另一個（基於身份的）秘密金鑰（解密金鑰）給使用者。解密時需要同時擁有兩把解密金鑰才可解密。免憑證密碼系統應用在電子簽名也是一樣，需有兩把簽名鑰匙才可簽名。因為PKG不能代替使用者解密或簽名，因此，解決了PKG的信賴性問題。另一方面，因為沒有金鑰憑證，所以沒有傳統方式的憑證註銷清單過大等問題。免憑證密碼系統因為不需驗證金鑰憑證，節省了許多複雜的計算，預計將可廣泛的應用於計算能力有限的電子機器像是手機或是PDA上。可是，免憑證密碼系統的研究至今仍屬於萌芽階段。截至目前為止，安全又有效率的免憑證密碼系統尚未存在，特別是在電子簽名的部分。舉例來說，在傳統電子簽名的研究部分，Boneh等[2]在2001年提出了簽名長度只需要160位元的短簽名。但是，安全的免憑證電子簽名至目前為止仍需至少320位元。Huang等[3]雖然在2006年提出了160位元的免憑證電子簽名，但犧牲了部分的安全性。因此，安全又有效率的免憑證密碼系統的研究至今仍是一個非常重要的課題。

利用身體特徵的公開金鑰密碼系統的研究動向

身體特徵指的是人體的生理特徵或行為上的特徵的訊息，一般可以根據臉型、指紋、掌紋、簽名以及眼睛及虹膜和聲音作為辨識依據，其中以指紋辨識的成本較低，利用也最廣。至目前為止，利用生體特徵的密碼系統主要利用在辨識使用者身份，譬如利用指紋登入或登出電腦系統或個人檔案的管理。另外，部分研究利用身體特徵與PKI結合，發展出利用身體特徵的加解密系統，此方式的缺點在於利用此系統之前，個人的身體特徵的訊息如指紋訊息等必須事先登入在PKI或管理者伺服器內，所以可能有侵犯個人隱私的問題或管理者的信賴問題。另外，因為指紋等等的身體特徵是永久不變的，當使用者退出系統時，如同現行PKI的憑證註銷問題一樣，如何註銷使用者的公開金鑰也是一個很大的問題。

研究方法



本計畫的著眼點首先在於免憑證公開金鑰密碼體系的二重金鑰的特徵上面。利用指紋等的身體特徵，嵌入二重金鑰中使用者自選的金鑰部分，然後研究並開發出安全又有效率的密碼系統。如同之前所介紹，本計畫將研究的免憑證公開金鑰密碼系統中，使用者的金鑰分成兩部分;CA所生成的（基於身份的）公開/秘密金鑰以及由使用者自己產生的公開/秘密金鑰。以使用者A為例。首先將A自己的姓名或E-mail住址發送給CA，由CA產生相對應的（基於身份的）秘密金鑰並回送給使用者A。使用者A將此秘密金鑰嵌入本計畫欲開發的裝置內，並利用指紋等的身體訊息加以加密保護。如此，欲生成A的電子簽名首先就必須要透過此裝置，要解密送給A的密文也必須要利用此裝置。除此之外，因為金鑰被利用者A的指紋訊息所加密保護，所以就算A的裝置被竊取了也不影響安全性（別人就算有此裝置也無法假冒A去做簽名或解密的行為）。然後，以指紋訊息作為免憑證密碼的第二把（秘密）金鑰並公開相對應的公開金鑰，如此一來，利用者A只需利用本計畫開發的裝置及自身的指紋訊息，就可輕鬆做出電子簽名或解密的行為。和PKI連結的方式不一樣的地方是，因為指紋等的個人隱私並沒有登入在PKI或管理者伺服器內，所以沒有個人隱私洩漏的問題，利用者退出系統時的金鑰註銷處理也可簡單完成。將人腦可記得的暗號相結合即可產生安全性更高的密碼系統。另一方面，因為CA不產生公開金鑰憑證，所以本研究的成果在簡化PKI上也是值得期待的。

執行步驟

在理論的研究上，首先從Pairing函數的研究開始，同時對免憑證公開金鑰密碼系統做廣泛及深入的研究。最終以基於指紋與通行碼(password)為加密或簽名鑰匙的免憑證密碼系統的實作與裝置開發為目標。

結果與討論

在計畫主持人及三位研究生的共同努力之下，本計畫最終達成了以下之成果。首先，在理論研究方面，本計畫是以提出新的免憑證密碼系統為目標。此部分共達成三項成果。

1. 提出了有效率及短簽名長度之免憑證簽名方案(Efficient and short certificateless signature)[7]，並發表於第七屆密碼與網路安全國際會議(7th International Conference on Crytology and Network Security)中。至目前為止，安全且有效率的免憑證公開金鑰密碼系統，尤其在電子簽名的部分尚未被提案出來。但是，要應用於計算能力有限的電子機器像是手機或是PDA上，安全與效率二者卻又是不可或缺的。本成果有效解決了前述所提之問題。
2. 提出具訊息回復功能之免憑證簽名方案(Certificateless signatures with message recovery) [8]。如同之前所介紹的，要將密碼系統有效應用於不具備複雜演算機能的機器如手機及PDA上，安全與效率是不可或缺的。訊息回復功能可減少訊息和簽章的總長度，提昇訊息的傳送效率。基於文獻[6]，我們提出了免憑證系統上的具訊息回復功能之免憑證簽名方案[8]。此成果已被2009全國計算機會議所

接受，預計於 11 月底於會議中發表。

3. 提出了一個新的的免憑證代理系統：代理簽章法(Proxy signature)的概念是在 1996 年由日本的學者 M.Mambo 提出[4]。在代理簽章的系統下，系統中的原簽章者可將簽章的動作賦予一代理簽章者執行。我們提出了一個新的代理簽章系統。此系統是基於免憑證的方式，所以較 Mambo 等提出的基於 PKI 架構下的方式更為有效率。另外，我們的方式可以很簡單的擴張為代理盲簽章系統。此系統中簽章者無法得知使用者欲簽名文件的內容，如此可以有效的保護使用者需要簽章的文件。本研究成果(Certificateless proxy signature and its extension to blind signature) [9] 已被 2009 全國計算機會議所接受，預計於 11 月底於會議中發表。

在系統的實做部分，透過本計畫的研究，並以理論研究方面的成果為基礎，本計畫最終實際開發製作出了基於指紋與通行碼(password)的免憑證密碼系統。此密碼系統不需複雜的操作，只需利用指紋就可實現加密或電子簽名的行為，如同現實世界中按壓指紋或蓋章的方式，因此任何人皆可輕鬆上手。另外，因為本計畫中的密碼是利用免憑證密碼系統，因此在簡化 PKI 上是值得期待的。此成果預計以專題的方式在系上展示。

計畫成果自評

本次研究內容與原申請計畫基本上是完全相符的。唯當初預計為二年期之計畫，但最終核可為一年期之計畫。因此一些理論部分的研究無法做得太深入，如 Map to point 函數的研究。此部分研究的目的是希望在系統實做時，能有更好的效率。除此之外，本計畫成果基本上已達成當初所預期之目標。理論研究上，共發表了三編文章。實做上也有了具體的成果。實做出之密碼系統不需複雜的操作，只需利用指紋就可實現加密或電子簽名的行為，如同現實世界中按壓指紋或蓋章的方式，因此任何人皆可輕鬆上手。另外，因為本計畫中的密碼是利用免憑證密碼系統，因此在簡化 PKI 上是值得期待的。

未來，除了對效率及安全性的繼續改良之外，也將檢討是否能有效應用於不具備複雜演算機能的機器如手機及 PDA 上，並做不同平台上之安全性與效率的評估。

相關文獻

[1] S.S. Al-Riyami, and K.G. Paterson, “Certificateless public key cryptography”, Advances in cryptology--ASISCRYPT’03, Lecture Notes in Computer Science Vol.2894, pp.452--473, 2003.

[2] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing”, Advances in cryptology --ASIACRYPT’01, Lecture Notes in Computer Science Vol.2248, pp.514--533, 2001.

[3] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, “Certificateles signature revisted”, In Proceedings of ACISP’07, Lecture Notes in Computer Science Vol.

4586, pp.308--322, 2007.

[4] M. Mambo, K. Usuda, and E. Okamoto, “Proxy signatures: Delegation of the power to sign messages”, IEICE Trans., 1996, E79-A, (9), pp. 1338-1354.

[5] A. Shamir, “Identity-based cryptosystems and signature schemes”, Advances in cryptology -- CRYPTO’84, Lecture Notes in Computer Science Vol.196, pp. 47—53, 1985.

[6] Raylin Tso, Chunxiang Gu, Takeshi Okamoto, and Eiji Okamoto, “Efficient ID-based digital signatures with message recovery”, in Proceedings of the 6th International Conference on Cryptology and Network Security (CANS2007), Springer, Lecture Notes in Computer Science, Vol. 4856, pp.47—59, 2007.

[7] Raylin Tso, Xun Yi, and Xinyi Huang, “Efficient and short certificateless signature”, in Proceedings of the 7th International Conference on Cryptology and Network Security (CANS2008), Lecture Notes in Computer Science, Vol. 5339, pp. 64--79, 2008. (本計畫相關之著作)

[8] 左瑞麟，詹省三，陳力瑋，陳淵順，”可訊息回復之免憑證簽章機制之研究”，2009 年全國計算機會議, accepted. (本計畫相關之著作)

[9] 左瑞麟，陳力瑋，陳淵順，詹省三，”免憑證代理簽名及其盲簽名之擴張”， 2009 年全國計算機會議, accepted. (本計畫相關之著作)

可供推廣之研發成果資料表

□ 可申請專利 ■ 可技術移轉 日期：98年10月25日

國科會補助計畫	計畫名稱：基於生物認證與暗號的免憑證密碼體系的研究與開發 計畫主持人： 左瑞麟 計畫編號： NSC － 97 － 2218 － E － 004 － 002 學門領域：E0818（資訊安全）
技術/創作名稱	兼具安全與操作便利性之加解密系統
發明人/創作人	左瑞麟
技術說明	中文：本技術利用指紋等的身體特徵，嵌入二重金鑰中使用者自選的金鑰部分，然後研究並開發出安全又有效率的密碼系統。在此系統中，使用者的金鑰分成兩部分;CA所生成的（基於身份的）公開/秘密金鑰以及由使用者自己產生的公開/秘密金鑰。以使用者A為例。首先將A自己的姓名或E-mail住址發送給CA，由CA產生相對應的（基於身份的）秘密金鑰並回送給使用者A。使用者A將此秘密金鑰嵌入本計畫欲開發的裝置內，並利用指紋等的身體訊息加以加密保護。如此，欲生成A的電子簽名首先就必須要透過此裝置，要解密送給A的密文也必須要利用此裝置。
	英文：The system of the device utilizes a user's bio-information (ie., the fingerprint) as his private key. In this way, the user does not have to remember a long and high entropy private key. In addition, to sign a document or to encrypt a message can be done very easily by just scan his fingerprint into our system (using the device we produced). This is very similar to take a person's fingerprint in real world and the user does not have to have the knowledge about cryptography.
可利用之產業 **及** **可開發之產品**	適用於以頻寬為主要考量之公司組織或產品如手機或PDA等。簡便的操作也適用於對電腦產品的操作不熟悉之使用者如年長或年幼者。
技術特點	因為金鑰被利用者A的指紋訊息所加密保護，所以就算A的裝置被竊取了也不影響安全性（別人就算有此裝置也無法假冒A去做簽名或解密的行為）。然後，以指紋訊息作為免憑證密碼的第二把（秘密）金鑰並公開相對應的公開金鑰，如此一來，利用者A只需利用本計畫開發的裝置及自身的指紋訊息，就可輕鬆做出電子簽名或解密的行為。

推廣及運用的價值	和 PKI 連結的方式不一樣的地方是，因為指紋等的個人隱私並沒有登入在 PKI 或管理者伺服器內，所以沒有個人隱私洩漏的問題，利用者退出系統時的金鑰註銷處理也可簡單完成。將人腦可記得的暗號相結合即可產生安全性更高的密碼系統。另一方面，因為 CA 不產生公開金鑰憑證，所以本研究的成果在簡化 PKI 上也是值得期待的。

※　1. 每項研發成果請填寫一式二份，一份隨成果報告送繳本會，一份送　貴單位研發成果推廣單位（如技術移轉中心）。

※　**2. 本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。**

※　3. 本表若不敷使用，請自行影印使用。

附錄: 發表論文

論文題目：Efficient and short certificateless signature
會議名稱：f the 7th International Conference on Cryptology and Network Security (CANS2008), Lecture Notes in Computer Science, Vol. 5339, pp. 64--79, 2008.

Efficient and Short Certificateless Signature

Raylin Tso[1,⋆], Xun Yi[2], and Xinyi Huang[3,⋆⋆]

[1] Department of Computer Science, National Chengchi University, Taiwan
raylin@cs.nccu.edu.tw
[2] School of Computer Science and Mathematics, Victoria University, Australia
Xun.Yi@vu.edu.au
[3] Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
xh068@uow.edu.au

Abstract. A certificateless signature (CLS) scheme with short signature size is proposed in this paper. Our scheme is as efficient as BLS short signature scheme in both communication and computation, and therefore turns out to be more efficient than other CLS schemes proposed so far. We provide a rigorous security proof of our scheme in the random oracle model. The security of our scheme is based on the k-CAA hard problem and a new discovered hard problem, namely, modified k-CAA problem. Our scheme can be applied to systems where signatures are typed in by human or systems with low-bandwidth channels and/or low-computation power, such as PDAs or cell phones.

Keywords: Bilinear pairing, certificateless signature, random oracle, short signature.

1 Introduction

Nowadays, the main difficulty in developing secure systems based on public key cryptography is the deployment and management of infrastructures to support the authenticity of cryptographic keys. The general approach to solve this problem is to use a Public Key Infrastructure (PKI) in which a trusted authority, called Certification Authority (CA), issues certificates to bind users and their public keys. However, the PKI is costly to use as it involves certificate revocation, storage, distribution, and verification.

In order to overcome the above mentioned problem, identity-based (ID-based) cryptography was firstly introduced by Shamir [19] in 1984. In an ID-based cryptosystem, one can use its unique identifier (e.g., names or e-mail addresses) as the public key. The user's identifier is publicly known and thus does not need certificates to prove its authenticity. Consequently, the problems associated with

⋆ Supported by National Science Council of Taiwan (NSC 97-2218-E-004-002).

⋆⋆ Supported by the National Natural Science Foundation of China (No. 60673070) and the Natural Science Foundation of Jiangsu Province (No. BK2006217).

M.K. Franklin, L.C.K. Hui, and D.S. Wong (Eds.): CANS 2008, LNCS 5339, pp. 64–79, 2008.

certificates can be eliminated. However, ID-based cryptosystems have an inherent key escrow issue as a third party "Private Key Generator" (PKG) generates the private keys for all users in the system. Therefore, the PKG must be fully trusted in ID-based cryptosystems.

Certificateless cryptography, firstly introduced by Al-Riyami and Paterson [2] in 2003, intends to solve the key escrow issue inherent in ID-based cryptography, and meanwhile to eliminate the use of certificates as in the conventional PKI. In a certificateless cryptosystem, private keys of users are generated by not only the PKG but also users themselves. In other words, PKG only issues a partial private key to each user while the user independently generates its additional public/secret key pair. Consequently, the PKG is unable to obtain secret keys of users. The cryptographic operations in certificateless system can be performed successfully only when both the partial private key and the secret key are known. In this way, the key escrow problem can be overcome. Following Al-Riyami and Paterson's pioneering work [2], many certificateless schemes have been proposed in recent years, such as [1,10,12,15,16,21,22,23,25] and etc..

1.1 Motivations

In the definition of the security model for certificateless signature (CLS) schemes, some papers (e.g., [1,14,15,17]) assume that the adversary should be allowed to obtain signatures signed with false public keys chosen by the adversary. But, in real world, the signatures that a "realistic" adversary can obtain are generated by a signer using the partial private key and the secret key corresponding to its original public key. Therefore, the adversary defined in those security models seems to enjoy more power than it could have in the real world. This assumption provides a higher security for the schemes on one hand but also limits the efficiency of the schemes on the other hand. This is because CLS schemes with a high security level usually sacrifice some efficiency in computation and/or communication and may not be practical for systems with low-bandwidth channels and/or low-computation power, such as PDAs or cell phones.

Except for the scheme proposed by Huang *et al.* [15], no secure CLS scheme has a short size of signature, although many short signatures in traditional PKI have been proposed [8,9,24]. As mentioned in [6], there are several important practical reasons for the desirableness of short signatures. For example, battery life is the major limitation on wireless devices such as PDAs, cell phones, RFID chips and sensors. Communicating even one bit of data uses significantly more power than executing one 32-bit instruction [3]. Reducing the number of bits to communicate saves power and is important to increase battery life. Also, in many settings, communication is not reliable, and thus the number of bits one has to communicate should be kept as few as possible. This inspired us to propose a more efficient certificateless short signature scheme.

1.2 Our Contributions

In this paper, on the basis of BLS short signature scheme [9], an efficient certificateless signature scheme with short signature size is proposed. Our scheme is as efficient as BLS short signature scheme (which is the traditional PKI model) in both communication and computation, and turns out to be more efficient than other CLS schemes proposed so far. This is achieved at the cost of stronger complexity assumptions.

In addition, as mentioned in [15], the security model defined in some CLS schemes (e.g., [16,21]) assume that, when an adversary queries the oracle **Public-Key-Replace** to replace a real public key with a false public key chosen by itself, the adversary is required to provide both the false public key and the corresponding secret value as the input. This is unreasonable since an adversary may pick a random public key for which the corresponding secret value is unknown even for himself. In other words, this definition may not cover the case in which an adversary may successfully forge a new signature with a false public key without knowing the corresponding secret value (to the false public key). Our definition for CLS scheme does not have such a problem and an adversary is not required to provide a secret value corresponding to a false public key as the input to the oracle **Public-Key-Replace**.

Based on the k-CAA problem, we define a new hard problem named "modified k-CAA problem". Assuming the hardness of these problems, we provide a rigorous security proof for our scheme in the random orale model .

The rest of this paper is organized as follows. In Section 2, we give some preliminaries (including the new discovered hardness assumption) which will be required throughout this paper. Section 3 is the presentation of our certificateless short signature scheme and in Section 4, we give the security proofs for our new scheme. Section 5 gives the performance comparison of our scheme with other schemes and the conclusion is given in Section 6.

2 Preliminaries

Before presenting our results, we first briefly review the notion of certificateless signature and its security definition. We will also review the definition for groups equipped with a bilinear map, and precisely state the hardness assumptions.

2.1 Certificateless Signatures

Following the definition in [2], a certificateless signature scheme is specified by seven randomized algorithms: **Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Sign** and **Verify**.

Setup. This algorithm takes as input a security parameter 1^k and returns the system parameters *params* and the master secret key *msk*. Usually, this algorithm is run by the KGC. We assume throughout that *params* are publicly and authentically available, but that only the KGC knows *msk*.

Partial-Private-Key-Extract. This algorithm takes the system parameter *params*, the master secret key *msk* and an identity *ID* as input. It returns a partial private key D_{ID}. Usually, this algorithm is run by the KGC and its output is transported to the identity *ID* over a confidential and authentic channel.

Set-Secret-Value. This algorithm takes as input the system parameter *params* and an identity *ID* as input and outputs a secret value x_{ID}. This algorithm is run by the identity *ID* for itself.

Set-Private-Key. This algorithm takes the system parameter *params*, a partial private key D_{ID} and a secret value x_{ID} of an identity *ID* as input. The value x_{ID} is used to transform D_{ID} into the (full) private key SK_{ID}. The algorithm returns SK_{ID}. This algorithm is run by the identity *ID* for itself.

Set-Public-Key. This algorithm takes the system parameter *params*, an identity *ID* and the identity's private key PK_{ID} as input. It outputs the public key PK_{ID} for the identity *ID*.

Sign. This algorithm takes the system parameter *params*, an identity *ID*, the private key SK_{ID} of *ID* and a message M as input. It outputs a certificateless signature σ.

Verify. This algorithm takes the system parameter *params*, an identity *ID*, the identity's public key PK_{ID} and a message/signature pair (M, σ) as input. It output *true* if the signature is correct, or $false$ otherwise.

2.2 Security Model

In this section, we discuss the definition of the security for a certificateless signature scheme.

For certificateless cryptosystems, the widely accepted notion of security was defined by Al-Riyami and Paterson in [2]. According to their definitions as well as the definitions in [25], there are two types of adversary with different capabilities:
Type I Adversary: This type of adversary $\mathcal{A}_I$ models a dishonest user who does not have access to the master key *msk* but has the ability to replace the public key of any entity with a value of his choice.
Type II Adversary: This type of adversary $\mathcal{A}_{II}$ models a malicious KGC who has access to the master key *msk* but *cannot perform public keys replacement*[1].

Generally, there are five oracles which can be accessed by the adversaries according to the game specifications which will be given later.

1. **Create-User:** On input an identity $ID \in \{0, 1\}^*$, if *ID* has already been created, nothing is to be carried out. Otherwise, the oracle runs the algorithms **Private-Key-Extract**, **Set-Secret-Value**, **Set-Public-Key** to obtain the partial private key D_{ID}, secret value x_{ID} and public key PK_{ID}. In this case, *ID* is said to be created. In both cases, PK_{ID} is returned.
2. **Public-Key-Replace:**[2] On input an identity *ID* and a user public key PK'_{ID}, the original user public key of *ID* is replaced with PK'_{ID} if *ID* has been created. Otherwise, no action will be taken.

[1] It is important that in certificateless cryptosystems, KGC must be semi-trusted and cannot perform the public key replacement. This is because that any adversary who knows the master key can impersonate anyone if he is allowed to replace the public key of the entity.

[2] Different from the security model defined in [16,21], in this oracle, an adversary is not required to provide the secret value x'_{ID} which is used to generated the public key PK'_{ID}.

3. **Secret-Value-Extract:** On input an identity, it returns the corresponding user secret key x_{ID} if ID has been created. Otherwise, returns a symbol $\perp$. Note that x_{ID} is the secret value associated with the original public key PK_{ID}. This oracle does not output the secret value associated with the replaced public key PK'_{ID}.
4. **Partial-private-Key-Extract:** On input an identity ID, it returns the partial private key D_{ID} if ID has been created. Otherwise, returns a symbol $\perp$.
5. **Sign:** On input an identity ID and a message $m \in \{0,1\}^*$, the signing oracle proceeds in one of the both cases below.
   - If ID has not been created, returns $\perp$.
   - If ID has been created, returns a valid signature σ such that $true \leftarrow \text{Veify}(m, \sigma, ID, PK_{ID})$. Here PK_{ID} is the public key returned from the oracle **Create-User**.

The standard notion of security for a signature scheme is called existential unforgeability against adaptive chosen message attack defined by Goldwasser, Micali and Revist [11]. To define the existential unforgeability of a certificateless signature against Type I adversary $\mathcal{A}_I$ and Type II adversary $\mathcal{A}_{II}$, we define two games, one for $\mathcal{A}_I$ and the other for $\mathcal{A}_{II}$.

Game 1: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_I$.

Setup. The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_I$ is given *params* but the master secret key *msk* is kept by the challenger.

Queries. $\mathcal{A}_I$ adaptively access all the oracles defined in Section 2.2 in a polynomial number of times.

Forgery. Eventually, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if the following conditions hold true:

1. $true \leftarrow \text{Verify}(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been submitted to the oracle **Sign**.
3. ID^* has never been submitted to the oracle **Partial-Private-Key-Extract** and **Secret-Value-Extract**.

Definition 1. Define $Adv_{\mathcal{A}_I}$ to be the probability that a Type I adaptively chosen message and chosen identity adversary $\mathcal{A}_I$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_I$ and the challenger. We say a certificateless signature scheme is secure against Type I attack, if, for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_I$, the success probability $Adv_{\mathcal{A}_I}$ is negligible.

Game 2: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_{II}$.

Setup. The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_{II}$ is given both *params* and *msk*.

Queries. $\mathcal{A}_{II}$ adaptively access all the oracles defined in Section 2.2 in a polynomial number of times.

Forgery. Eventually, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if the following conditions hold true:

1. $true \leftarrow \text{Verify}(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been queried to the oracle **Sign**.
3. ID^* has never been submitted to the oracle **Secret-Value-Extract**.

Definition 2. Define $Adv_{\mathcal{A}_{II}}$ to be the probability that a Type II adaptively chosen message and chosen identity adversary $\mathcal{A}_{II}$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_{II}$ and the challenger. We say a certificateless signature scheme is secure against Type II attack, if, for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{II}$, the success probability $Adv_{\mathcal{A}_{II}}$ is negligible.

Definition 3. A certificateless signature scheme is existentially unforgeable against adaptive chosen message and chosen identity attack if it is secure against both Type I and Type II attacks defined above.

2.3 Bilinear Groups and Complexity Assumptions

Let $\mathbb{G}_1, \mathbb{G}_2$ be two multiplicative cyclic groups of order p for some large prime p. Our scheme makes use of the *bilinear* map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between these two groups. The bilinear map should be satisfied with the following properties:

1. **Bilinear:** A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$ for all $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$.
2. **Non-degenerate:** The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$. Observe that since $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order, this implies that if g is a generator of $\mathbb{G}_1$, then $\hat{e}(g, g)$ is a generator of $\mathbb{G}_2$.
3. **Computable:** There is an efficient algorithm to compute $\hat{e}(g, h)$ for any $g, h \in \mathbb{G}_1$.

A bilinear map satisfying the three properties above is said to be an *admissible* bilinear map. We can make this map using the Weil pairing or the Tate pairing [4,5,9]

Next, we describe the complexity assumptions which are required for the security proof of our scheme.

We first introduce a problem given by Mitsunari *et al.* [18] which is called k-CAA (Collusion Attack Algorithm with k traitors) problem and then give a modified problem.

Definition 4. **k-CAA Problem** [18]
For $x, h_1, \cdots, h_k \in \mathbb{Z}_p^*$, and a generator g of $\mathbb{G}_1$. Given g, g^x and k pairs $(h_1, g^{(x+h_1)^{-1}}), \cdots, (h_k, g^{(x+h_k)^{-1}})$, output a new pair $(h^*, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$.

The k-CAA problem is believed to be hard. Mitsunari *et al.* firstly introduced this problem and gave a traitor tracing scheme [18] based on this problem. Although their application to tracing traitors is proved by Tô *et al.* [20] to be insecure, the k-CCA problem still remains to be hard without broken. Zhang et al. [24] recently gave a secure and efficient signature scheme based on the same problem.

In addition to the k-CAA problem, the security of our scheme also bases on a modified version of the original k-CAA problem. We call it as the Modified k-CAA Problem which is defined as follows:

Definition 5. Modified k-CAA Problem
For randomly picked $x, a, b, h_1, \cdots, h_k \in \mathbb{Z}_p^*$, and a generator g of $\mathbb{G}_1$. Let $g_1 = g^{ab} \neq g$. Given g, g^x, g^a, g^b, g^{bx} and k pairs $(h_1, g_1^{(x+h_1)^{-1}}), \cdots, (h_k, g_1^{(x+h_k)^{-1}})$, output either a new pair $(h^*, g_1^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$ or g_1.

Note that in the above definition, g_1 is not given to the problem. If we define $g_1 = g$ in the input, then g^a, g^b and g^{bx} are useless and can be ignored. In this case, the problem is to find a new pair $(h^*, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$.

3 The Proposed Certificateless Short Signature Scheme

In this section, we will describe our certificateless short signature scheme. It consists of the following algorithms:

Setup: Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups of some prime order $p \geq 2^k$, k be the security parameter of the scheme. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is an admissible bilinear pairing. Let $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be two secure cryptographic hash functions. KGC chooses a random number $s \in \mathbb{Z}_p^*$ and an arbitrary generator $g \in \mathbb{G}_1$. It sets $P_{pub} = g^s$, publishes $params = \{\mathbb{G}_1, \mathbb{G}_2, g, \hat{e}, H_0, H_1, P_{pub}\}$ and keeps the master secret key $msk = s$ secretly.

Partial-Private-Key-Extract: Given an entity's identity $ID \in \{0, 1\}^*$, KGC sets $Q_{ID} = H_0(ID)$ and computes the entity's partial private key $D_{ID} = Q_{ID}^s$. KGC transmits D_{ID} to ID over a confidential and authentic channel.

Set-Secret-Value: The entity ID chooses a random number $x_{ID} \in \mathbb{Z}_p^*$.

Set-Private-Key: The entity ID sets his private key as $SK_{ID} = (D_{ID}, x_{ID})$.

Set-Public-Key: Given x_{ID}, the entity ID computes the public key $PK_{ID} = (PK_1, PK_2) = (g^{x_{ID}}, Q_{ID}^{x_{ID}})$.

Sign: To sign a message $m \in \{0,1\}^*$, the entity ID first sets $h = H_1(m||ID||PK_{ID})$ and then computes the signature $\sigma = D_{ID}^{(x_{ID}+h)^{-1}}$.

Verify: Given a pair (m, σ) and ID's public key $PK_{ID} = (g^{x_{ID}}, Q_{ID}^{x_{ID}})$, any verifier first checks the equation $\hat{e}(PK_1, Q_{ID}) = \hat{e}(PK_2, g)$. If it holds, then computes $h = H_1(m||ID||PK_{ID})$ and checks the equation

$$\hat{e}(\sigma, PK_1 \cdot g^h) \stackrel{?}{=} \hat{e}(H_0(ID), P_{pub}).$$

If the equality holds, outputs $true$, otherwise, outputs $false$.

Correctness: If σ is a valid signature on m, then the correctness holds since

$$\begin{aligned}&\hat{e}(\sigma, PK_1 \cdot g^h)\\&= \hat{e}(D_{ID}^{(x_{ID}+h)^{-1}}, g^{x_{ID}} \cdot g^h) = \hat{e}(H_0(ID)^{s(x_{ID}+h)^{-1}}, g^{x_{ID}+h})\\&= \hat{e}(H_0(ID), g)^{s(x_{ID}+h)^{-1}(x_{ID}+h)} = \hat{e}(H_0(ID), g)^s\\&= \hat{e}(H_0(ID), g^s) = \hat{e}(H_0(ID), P_{pub}).\end{aligned}$$

4 Security Proofs

Theorem 1. Unforgeability against Type I Adversary: If there exists a Type I adaptively chosen message and chosen *ID* adversary $\mathcal{A}_I$ who can ask at most q_C **Create-User** queries, q_{KEx} **Partial-Private-Key-Extract** queries, q_{VEx} **Secret-Value-Extract** queries and q_S **sign** queries, respectively, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, using $\mathcal{A}_I$ as a black box, can solve the modified k-CAA problem [Definition 5] (where $k \geq q_S$ and is in proportion to the number of the H_1-hash queries) with probability $Adv_{\mathcal{F}}^{mk-CAA} \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1 - \frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon$.

Proof: If there exists an adversary $\mathcal{A}_I$ who can break the unforgeability of the proposed scheme via Type I attack, then, we can construct another adversary $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_I$ as a black-box and solve the modified k-CAA problem.

Let g be a generator of $\mathbb{G}_1$, x, a, b be three random numbers of $\mathbb{Z}_p^*$ and $g_1 = g^{ab} \in \mathbb{G}_1$. Let $h_1, \cdots, h_k \in \mathbb{Z}_p^*$ be k random numbers. $\mathcal{F}$ is given the challenge $\{g, g^x, g^a, g^b, g^{bx}, (h_1, g_1^{(x+h_1)^{-1}}), \cdots, (h_k, g_1^{(x+h_k)^{-1}})\}$. The purpose of $\mathcal{F}$ is either to find a new pair $(h^*, g_1^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$ or to find g_1, which are the solutions to the modified k-CAA problem.

Setup: In order to solve the problem, $\mathcal{F}$ utilizes $\mathcal{A}_I$ as a black-box. To get the black-box $\mathcal{A}_I$ run properly, $\mathcal{F}$ will simulate the environments of the proposed scheme and the oracles which $\mathcal{A}_I$ can access. In this proof, we regard the hash functions H_0, H_1 as random oracles. $\mathcal{F}$ starts by picking an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and sets $P_{pub} = g^a$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ to $\mathcal{A}_I$ and allows $\mathcal{A}_I$ to run.

Due to the ideal randomness of the H_1-hash, we may assume that $\mathcal{A}_I$ is well-behaved in the sense that it always requests a H_1-hash of $m||ID||PK_{ID}$ before it requests a signature for m signed by ID's public key PK_{ID}. In addition, it always requests a H_1-hash of $m^*||ID^*||PK_{ID^*}$ that it outputs as its forgery. It is trivial to modify any adversary-algorithm $\mathcal{A}_I$ to have this property.

Query: At any time, $\mathcal{A}_I$ is allowed to access the following oracles in a polynomial number of times. These oracles are all simulated by $\mathcal{F}$.

1. **Create-User:** $\mathcal{A}_I$ can query this oracle by given an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \cdots, q_C\}$.

   (1) If $i \neq t$, $\mathcal{F}$ chooses $d_i, x_i \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_i) = g^{d_i}$, $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)}) = (g^{x_i}, g^{d_i x_i})$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = H_0(ID_i)^a = g^{ad_i} = P_{pub}^{d_i}$ and the secret value is $x_{ID_i} = x_i$.

   (2) If $i = t$, $\mathcal{F}$ sets $H_0(ID_t) = g^b$ and $PK_{ID_t} = (PK_{(ID_t,1)}, PK_{(ID_t,2)}) = (g^x, g^{bx})$. In this case, $\mathcal{F}$ will set $D_{ID_t} = x_{ID_t} = \perp$ which means that it cannot compute the secret value and the partial private key of ID_t.

   In both cases, returns $H_0(ID_i)$ and PK_{ID_i}.

2. **Partial-Private-Key-Extract:** At any time, $\mathcal{A}_I$ can query the oracle by given an identity ID_i. $\mathcal{F}$ outputs a symbol $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns $D_{ID_i} = g^{ad_i}$. Otherwise, $\mathcal{F}$ returns *failure* and terminates the simulation.
3. **Public-Key-Replace:** $\mathcal{A}_I$ can request to replace public key PK_{ID_i} of an entity ID_i with new public key PK'_{ID_i} chosen by $\mathcal{A}_I$ itself. $\mathcal{F}$ replaces the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, outputs $\perp$. Here, to replace a public key, the secret value corresponding to the new public key is not required.
4. **Secret-Value-Extract:** Given ID_i chosen by $\mathcal{A}_I$, outputs $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns x_{ID_i} to $\mathcal{A}_I$. Otherwise, $i = t$ and $\mathcal{F}$ reports *failure* and terminates the simulation.
5. H_1 **Queries:** $\mathcal{A}_I$ can query the random oracle H_1 at any time on an input $\omega_i = (m_l || ID_j || PK_{ID_k})$. For i-th H_1 query asked by $\mathcal{A}_I$ on input ω_i, $\mathcal{F}$ first checks if $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$ or not. Here PK_{ID_t} is the original public key.
   - If $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$, then $\mathcal{F}$ first flips a biased coin which outputs a value $c_i = 1$ with probability ζ, and $c_i = 0$ with probability $1 - \zeta$ (the value of ζ will be optimized later).
     (1) If $c_i = 1$, $\mathcal{F}$ picks a random value $h'_i \in \mathbb{Z}_p^*$ where $h'_i \notin \{h_1, \cdots, h_k\}$ and responds h'_i to $\mathcal{A}_I$ as the value of $H_1(\omega_i)$.
     (2) If $c_i = 0$, $\mathcal{F}$ returns a value $h''_i \in_R \{h_1, \cdots, h_k\}$ as the output of $H_1(\omega_i)$ where h''_i must be a fresh value which means that it has not been assigned as an output of H_1 queries before.
   - Otherwise, $\mathcal{F}$ picks and responds with a random value $\mu_i \in \mathbb{Z}_p^*$.

   In either cases, $\mathcal{F}$ records (ω_i, h'_i, c_i), (ω_i, h''_i, c_i) or (ω_i, μ_i) to a H_1-*List* which is initially empty.
6. **Sign:** For each sign query on an input (m_l, ID_j), output $\perp$ if ID_j has not been created. For any input (m_l, ID_j) with ID_j which has already been created, since we assume that $\mathcal{A}_I$ is well-behaved, we know that $\mathcal{A}_I$ has already queried the random oracle H_1 on the input $\omega_i = (m_l || ID_j || PK_{ID_j})$.
   - If $ID_j \neq ID_t$, $\mathcal{F}$ uses the private key (x_{ID_j}, D_{ID_j}) of ID_j and $\mu_i = H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_l and the identity ID_j.
   - If $ID_i = ID_t$, then, $\mathcal{F}$ first checks the H_1-*List*.
     (1) If $c_i = 1$, $\mathcal{F}$ reports *failure* and terminates the simulation.
     (2) Otherwise, $c_i = 0$ and $h''_i = H_1(m_l || ID_t || PK_{ID_t})$ is on the H_1-*List*. For easy of description, we assume $h''_i = h_i \in \{h_1, \cdots, h_k\}$.

$\mathcal{F}$ then returns $\sigma_i = g_1^{(x+h_i)^{-1}}$. Note that

$$\hat{e}(\sigma_i, PK_{(ID_t,1)} \cdot g^{h_i}) = \hat{e}(g_1^{(x+h_i)^{-1}}, g^x \cdot g^{h_i}) = \hat{e}(g_1, g)$$
$$= \hat{e}(g^{ab}, g) = \hat{e}(g^b, g^a) = \hat{e}(H_0(ID_t), P_{pub}).$$

Therefore, σ_i is a valid signature on m_l and ID_t.

Forgery: After all the queries, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK_{ID^*} = (PK_{(ID^*,1)}, PK_{(ID^*,2)}), m^*, \sigma^*)$ and wins the game.
If σ^* is a valid forgery, then $h^* = H_1(m^*||ID^*||PK_{ID^*})$ which is on the H_1-*List*, and $\hat{e}(\sigma^*, PK_{(ID^*,1)} \cdot g^{h^*}) = \hat{e}(H_0(ID^*), P_{pub})$ where $PK_{ID^*} = g^{x^*}$ may be a new public key replaced by $\mathcal{A}_I$ or the original public key generated by the oracle **Create-User**. In addition, $\hat{e}(PK_{(ID^*,1)}, Q_{ID^*}) = \hat{e}(PK_{(ID^*,2)}, g)$ if $\mathcal{A}_I$ wins the game. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs *failure* and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.
(1) If $c^* = 0$, $\mathcal{F}$ outputs *failure* and terminates the simulation.
(2) Otherwise, $c^* = 1$ and $h^* \notin \{h_1, \cdots, h_k\}$. If $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = (PK_{(ID_t,1)}, PK_{(ID_t,2)})$ is the original public key generated by the oracle, then, $\mathcal{F}$ outputs a new pair $(h^*, \sigma^*) = (h^*, g_1^{(x+h^*)^{-1}})$ which will be the solution to the modified k-CAA problem. If $(PK_{(ID^*,1)}, PK_{(ID^*,2)})$ is a new public key replaced by $\mathcal{A}_I$, then, using the knowledge of exponent assumption introduced in [7,13], $\mathcal{F}$ can either extract x^* if $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = (g^*, g^{bx^*})$ is generated from (g, g^b) or extract r if $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = ((g^x)^r, (g^{bx})^r)$ is generated from (g^x, g^{bx}). Consequently, $g_1 = (\sigma^*)^{(x^*+h^*)}$ can be computed if x^* extracted or a new pair $(h', g_1^{(x+h')^{-1}}) = (h^*/r, (\sigma^*)^r)$ can be found if r extracted, which is also the solution to the modified K-CAA problem.

It remains to compute the probability that $\mathcal{F}$ solves the modified k-CAA problem. Actually, $\mathcal{F}$ succeeds if:

Λ_1 : $\mathcal{F}$ does not abort during the simulation.
Λ_1 : σ^* is a valid forgery on (ID^*, PK_{ID^*}, m^*).
Λ_1 : $ID^* = ID_t$ and $c^* = 1$.

The advantage of $\mathcal{F}$ is $Adv_{\mathcal{F}}^{BCk-CAA} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2]$. If Λ_1 happens, then:

- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Partial-Private-Key-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{PKEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{VEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of **sign** oracle. This happens with probability $(1 - \frac{1}{q_C}\zeta)^{q_S} \geq (1 - \zeta)^{q_S}$.

Consequently, $Pr[\Lambda_1] \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1 - \zeta)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \frac{\zeta}{q_C}$. Therefore, $Adv_{\mathcal{F}}^{BCk-CAA} \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}$

Table 1. Performance Evaluation

	Type	PK-Size (bits)	Sig-Length (bits)	Sign	Verify
Ours	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$1\hat{e} + 1E_{G_1}$
BLS [9]	No	$\|G_1\|$ (≈ 160)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$1\hat{e} + 1E_{G_1}$
Scheme I [15]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$2\hat{e}$
ZWXF [25]	CLS	$\|G_1\|$ (≈ 160)	$2\|G_1\|$ (≈ 320)	$3E_{G_1}$	$3\hat{e}$
HSMZ [16]	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\| + 1\|q\|$ (≈ 320)	$2E_{G_1} + 1E_{G_2}$	$1\hat{e} + 1E_{G_2}$
AP03 [2]	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\| + 1\|q\|$ (≈ 320)	$2E_{G_1} + 1E_{G_2}$	$1\hat{e} + 1E_{G_2}$
eCLS [10]	CLS	$\|G_1\|$ (≈ 160)	$2\|G_1\|$ (≈ 320)	$2E_{G_1}$	$2\hat{e} + 1E_{G_1}$
Scheme II [15]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\| + 2\|p\|$ (≈ 480)	$3E_{G_1} + 1E_{G_2}$	$1\hat{e} + 3E_{G_1} + 1E_{G_2}$
oCLS [10]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\| + \|G_2\|$ (≈ 1184)	$2E_{G_1}$	$1\hat{e} + 1E_{G_2}$

$(1-\zeta)^{q_S}\frac{\zeta}{q_C}\varepsilon$. The function $\zeta(1-\zeta)^{q_S}$ is maximized at $\zeta = \frac{1}{q_S+1}$. Therefore,

$$Adv_{\mathcal{F}}^{BCk-CAA} \geq (1-\frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1-\frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon.$$

This ends the proof. □

Theorem 2. Unforgeability against Type II Adversary: If there exists a Type II adaptively chosen message and chosen *ID* adversary $\mathcal{A}_{II}$ who can ask at most q_C **Create-User** queries, q_{VEx} **Secret-Value-Extract** queries and q_S **Sign** queries, respectively, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, using $\mathcal{A}_{II}$ as a black box, can solve the k-CAA problem [Definition 4] (where $k \geq q_S$ and is in proportion to the number of the H_1-hash queries) with probability $Adv_{\mathcal{F}}^{k_CAA} \geq (1-\frac{1}{q_C})^{q_{VEx}}(1-\frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon$.

Proof: The proof is similar to that of proving Theorem 1 with a little modification. See Appendix for details. □

Theorem 1 is proved in a relatively weaker model than the normal one. That is, we do not allow the adversary to obtain valid signatures according to the replaced public key.

As mentioned in Section 1, this model is also acceptable as the signatures that a "realistic" adversary can obtain are usually generated by a signer under its original public key. Therefore, this modification is reasonable and Huang et al.'s first scheme with short signature size [15] is also analyzed in this weak model.

5 Performance Comparison

In this section, we compare our certificateless short signature scheme with other existing CLS schemes and BLS short signature scheme [9] from the aspect of communication cost and computation cost in signature signing and verification, respectively.

In the comparison, the operations such as $\hat{e}(g, g)$, $\hat{e}(PK_1, Q_{ID}) = \hat{e}(PK_2, g)$ or $\hat{e}(H_0(ID), P_{pub})$ are pre-computable or only need to be computed once. Therefore, these computations are neglected in the comparison. In Table 1, certificateless signature schemes are marked with "CLS". Other schemes are marked with"No". We denote by $\hat{e}$ a computation of the pairing, E_{G_1} an exponentiation in $\mathbb{G}_1$, and E_{G_2} an exponentiation in $\mathbb{G}_2$. Usually, pairing operations cost much more than other computations. One $\hat{e}$ operation is about 10 times more expensive than one $E_{(.)}$ operation.

We can see in Table 1 that our scheme is as efficient as BLS short signature [9] but our scheme is certificateless whereas BLS scheme is not. This means there is no need to verify a certificate in our scheme while using BLS scheme, a verifier needs to verify the certificate in order to confirm the correctness of the public key, as in the conventional Public key Infrastructure (PKI), which is generally considered to be costly to use and manage. From this point of view, our scheme is superior than BLS short signature scheme.

Among all certificateless signature schemes, Huang et al.'s first scheme in [15] is the only signature scheme providing short signature-length (about 160 bits) as ours. However, our scheme is more efficient than their scheme in the verification phase. To the best of our knowledge, our scheme is the most efficient CLS scheme in the aspects of both communication and computation costs.

6 Conclusion

In this paper, we proposed a certificateless signature scheme which is as efficient as BLS short signature. We also defined a new hard problem "modified k-CAA problem" based on the k-CAA problem. The security of the proposed scheme is proved in the random oracle model under the hardness of k-CAA problem and modified k-CAA problem.

References

1. Au, M.H., Chen, J., Liu, J.K., Mu, Y., Wong, D.S., Yang, G.: Malicious KGC attacks in certificateless cryptography. In: Proceedings of ASIACCS 2007, pp. 302–311 (2007)
2. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
3. Barr, K., Asanovic, K.: Energy aware lossless data compression. In: Proceedings of the ACM Conference on Mobile Systems, Applications, and Services (MobiSys) (2003)
4. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithm for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–369. Springer, Heidelberg (2002)
5. Barreto, P.S.L.M., Lynn, B., Scott, M.: On the selection of pairing-friendly groups. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 17–25. Springer, Heidelberg (2004)

6. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM Confetence on Computer and Communication Security, pp. 390–398 (2006)
7. Bellare, M., Palacio, A.: The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)
8. Boneh, D., Boyen, X.: Short signatures withou rando oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
9. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–533. Springer, Heidelberg (2001)
10. Choi, K.Y., Park, J.H., Hwang, J.Y., Lee, D.H.: Efficient certificateless signature schemes. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 443–458. Springer, Heidelberg (2007)
11. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal of Computing 17(2), 281–308 (1988)
12. Gorantla, M.C., Saxena, A.: An efficient certificateless signature scheme. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS, vol. 3802(II), pp. 110–116. Springer, Heidelberg (2005)
13. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 408–423. Springer, Heidelberg (1998)
14. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Certificatelss signature: a new security model and an improved generic construction. Designs, Codes and Cryptography 42(2), 109–126 (2007)
15. Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless signature revisted. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 308–322. Springer, Heidelberg (2007)
16. Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the security of certificateless signature schemes from Asiacrypt 2003. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (eds.) CANS 2005. LNCS, vol. 3810, pp. 13–25. Springer, Heidelberg (2005)
17. Liu, J.K., Au, M.H., Susilo, W.: Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: Proceedings of ASIACCS 2007, pp. 273–283 (2007)
18. Mitsunari, S., Sakai, R., Kasahara, M.: A new traitor tracing. Journal of IEICE Trans. Fundamentals E85-A(2), 481–484 (2002)
19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
20. Tô, V., Safavi-Naini, R., Zhang, F.: New traitor tracing schemes using bilinear map. In: Proceedings of 2003 DRM Workshop, pp. 67–76 (2003)
21. Yap, W.L., Heng, S.H., Goi, B.M.: An efficient certificteless signature. In: Zhou, X., Sokolsky, O., Yan, L., Jung, E.-S., Shao, Z., Mu, Y., Lee, D.C., Kim, D.Y., Jeong, Y.-S., Xu, C.-Z. (eds.) EUC Workshops 2006. LNCS, vol. 4097, pp. 322–331. Springer, Heidelberg (2006)
22. Yap, W.L., Chow, S.S.M., Heng, S.H., Goi, B.M.: Security Mediated Certificateless Signatures. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 459–477. Springer, Heidelberg (2007)

23. Yum, D.H., Lee, P.J.: Generic construction of certificateless signature. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 200–211. Springer, Heidelberg (2004)
24. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from binilear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004)
25. Zhang, Z., Wong, D.S., Xu, J., Feng, D.: Certificateless public-key signature: security model and efficiet construction. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 293–308. Springer, Heidelberg (2006)

Appendix

Proof of Theorem 2

Proof: If there exists an adversary $\mathcal{A}_{II}$ who can break the unforgeability of the proposed scheme via Type II attack, then, we can construct another adversary $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_{II}$ as a black-box and solve the k-CCA problem.

Let g be a generator of $\mathbb{G}_1$, and $x, h_1, \cdots, h_k \in \mathbb{Z}_p^*$ be $k+1$ random numbers. $\mathcal{F}$ is given the challenge $\{g, g^x, (h_1, g^{(x+h_1)^{-1}}), \cdots, (h_k, g^{(x+h_k)^{-1}})\}$. The purpose of $\mathcal{F}$ is to output a tuple $(h, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$, which is the solution to the k-CAA problem.

Setup: In order to solve the problem, $\mathcal{F}$ utilizes $\mathcal{A}_{II}$ as a black-box. To get the black-box $\mathcal{A}_{II}$ run properly, $\mathcal{F}$ will simulate the environments of the proposed scheme and the oracles which $\mathcal{A}_{II}$ can access. In this proof, we regard the hash functions H_0, H_1 as random oracles. $\mathcal{F}$ starts by picking an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and sets $P_{pub} = g^s$, where s is randomly chosen from $\mathbb{Z}_p^*$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ together with the master secret key s to $\mathcal{A}_{II}$ and allows $\mathcal{A}_{II}$ to run.

Due to the ideal randomness of the H_1-hash, we may assume that $\mathcal{A}_{II}$ is well-behaved in the sense that it always requests a H_1-hash of $m||ID||PK_{ID}$ before it requests a signature for m signed by ID's public key PK_{ID}. In addition, it always requests a H_1-hash of $m^*||ID^*||PK_{ID^*}$ that it outputs as its forgery. It is trivial to modify any adversary-algorithm $\mathcal{A}_{II}$ to have this property.

Query: At any time, $\mathcal{A}_{II}$ is allowed to access the following oracles in a polynomial number of times. These oracles are all simulated by $\mathcal{F}$. Different from the proof for Type I adversary, there is no oracle **Partial-Private-Key-Extract**. This is because that $\mathcal{A}_{II}$ has already obtained the master secret key s so he can compute the partial private key (i.e., $D_{ID} = H_0(ID)^s$)) of any entity using the master key s.

1. **Create-User:** $\mathcal{A}_{II}$ can query this oracle by given an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \cdots, q_C\}$.

   (1) If $i \neq t$, $\mathcal{F}$ chooses $d_i, x_i \in_R \mathbb{Z}_p^*$ and computes $H_0(ID_i) = g^{d_i}$, $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)}) = (g^{x_i}, g^{x_i d_i})$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = g^{s d_i}$ and the secret value is $x_{ID_i} = x_i$.

(2) If $i = t$, $\mathcal{F}$ chooses $d_t \in_R \mathbb{Z}_p^*$ and computes $H_0(ID_t) = g^{d_t}$. However, $\mathcal{F}$ sets $PK_{ID_t} = (PK_{(ID_t,1)}, PK_{(ID_t,2)}) = (g^x, g^{xd_t})$. In this case, $\mathcal{F}$ will set $D_{ID_t} = g^{sd_t}$ and $x_{ID_t} = \perp$ which means that it cannot compute the secret value of ID_t.

In both cases, returns $H_0(ID_i)$ and PK_{ID_i}.

2. **Public-Key-Replace:** $\mathcal{A}_{II}$ can request to replace public key PK_{ID_i} of an entity ID_i with new public key PK'_{ID_i} chosen by $\mathcal{A}_{II}$ itself. $\mathcal{F}$ replaces the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, outputs $\perp$. Here, to replace a public key, the secret value corresponding to the new public key is not required.
3. **Secret-Value-Extract:** Given ID_i chosen by $\mathcal{A}_{II}$, outputs $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns x_{ID_i} to $\mathcal{A}_{II}$. Otherwise, $i = t$ and $\mathcal{F}$ reports *failure* and terminates the simulation.
4. H_1 **queries:** $\mathcal{A}_{II}$ can query the random oracle H_1 at any time on an input $\omega_i = (m_l||ID_j||PK_{ID_k})$. For i-th H_1 query asked by $\mathcal{A}_{II}$ on input ω_i, $\mathcal{F}$ first checks if $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$ or not. Here PK_{ID_t} is the original public key.
   - If $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$, then $\mathcal{F}$ first flips a biased coin which outputs a value $c_i = 1$ with probability ζ, and $c_i = 0$ with probability $1 - \zeta$ (the value of ζ will be optimized later).
     (1) If $c_i = 1$, $\mathcal{F}$ picks a random value $h'_i \in \mathbb{Z}_p^*$ where $h'_i \notin \{h_1, \cdots, h_k\}$ and responds h'_i to $\mathcal{A}_{II}$ as the value of $H_1(\omega_i)$.
     (2) If $c_i = 0$, $\mathcal{F}$ returns a value $h''_i \leftarrow_R \{h_1, \cdots, h_k\}$ as the output of $H_1(\omega_i)$ where h''_i must be a fresh value which means that it has not been assigned as an output of H_1 queries before.
   - Otherwise, $\mathcal{F}$ picks and responds with a random value $\mu_i \in \mathbb{Z}_p^*$.

   In either cases, $\mathcal{F}$ records (ω_i, h'_i, c_i), (ω_i, h''_i, c_i) or (ω_i, μ_i) to a H_1-*List* which is initially empty.
5. **Sign:** For each sign query on an input (m_l, ID_j), output $\perp$ if ID_j has not been created. For any input (m_l, ID_j) with ID_j which has already been created, since we assume that $\mathcal{A}_{II}$ is well-behaved, we know that $\mathcal{A}_{II}$ has already queried the random oracle H_1 on the input $\omega_i = (m_l||ID_j||PK_{ID_j})$.
   - If $ID_j \neq ID_t$, $\mathcal{F}$ uses the private key (x_{ID_j}, D_{ID_j}) of ID_j and $\mu_i = H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_l and the identity ID_j.
   - If $ID_i = ID_t$, then, $\mathcal{F}$ first checks the H_1-*List*.
     (1) If $c_i = 1$, $\mathcal{F}$ reports *failure* and terminates the simulation.
     (2) Otherwise, $c_i = 0$ and $h''_i = H_1(m_l||ID_t||PK_{ID_t})$ is on the H_1-*List*. For easy of description, we assume $h''_i = h_i \in \{h_1, \cdots, h_k\}$. $\mathcal{F}$ then returns $\sigma_i = g^{sd_t(x+h_i)^{-1}}$. Note that

$$\begin{aligned}\hat{e}(\sigma_i, PK_{(ID_t,1)} \cdot g^{h_i}) &= \hat{e}(g^{sd_t(x+h_i)^{-1}}, g^x \cdot g^{h_i}) = \hat{e}(g^{sd_t}, g)\\ &= \hat{e}(g,g)^{sd_t} = \hat{e}(g^{d_t}, g^s) = \hat{e}(H_0(ID_t), P_{pub}).\end{aligned}$$

     Therefore, σ_i is a valid signature on m_l and ID_t.

Forgery: After all the queries, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK_{ID^*} = (PK_{(ID^*,1)}, PK_{(ID^*,2)}), m^*, \sigma^*)$ and wins the game.
If σ^* is a valid forgery, then $h^* = H_1(m^*||ID^*||PK_{ID^*})$ which is on the H_1-*List*, and

$$\hat{e}(\sigma^*, PK_{(ID^*,1)} \cdot g^{h^*}) = \hat{e}(H_0(ID^*), P_{pub})$$

where $PK_{(ID^*,1)} = g^{x_{ID^*}}$ must be the original public key generated by the oracle **Create-User**. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs *failure* and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.

(1) If $c^* = 0$, $\mathcal{F}$ outputs *failure* and terminates the simulation.
(2) Otherwise, $c^* = 1$ and $h^* \notin \{h_1 \cdots, h_k\}$. $\mathcal{F}$ computes $\xi = (\sigma^*)^{(sd_t)^{-1}}$ and outputs the tuple $(h^*, \xi) = (h^*, g^{(x+h^*)^{-1}})$ which will be the solution to the k-CAA problem.

It remains to compute the probability that $\mathcal{F}$ solves the k-CAA problem. Actually, $\mathcal{F}$ succeeds if:

$\Lambda_1 :$ $\mathcal{F}$ does not abort during the simulation.
$\Lambda_2 :$ σ^* is a valid forgery on (ID^*, PK_{ID^*}, m^*).
$\Lambda_3 :$ $ID^* = ID_t$ and $c^* = 1$.

The advantage of $\mathcal{F}$ is

$$Adv_{\mathcal{F}}^{k-CAA} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2].$$

If Λ_1 happens, then

- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{VEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of signing oracle. This happens with probability $(1 - \frac{1}{q_C}\zeta)^{q_S} \geq (1 - \zeta)^{q_S}$.

Consequently, $Pr[\Lambda_1] \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1 - \zeta)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \frac{\zeta}{q_C}$. Therefore, $Adv_{\mathcal{F}}^{k-CAA} \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1 - \zeta)^{q_S}\frac{\zeta}{q_C}\varepsilon$. The function $\zeta(1 - \zeta)^{q_S}$ is maximized at $\zeta = \frac{1}{q_S+1}$. Therefore,

$$Adv_{\mathcal{F}}^{k-CAA} \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1 - \frac{1}{q_S + 1})^{q_S}\frac{1}{q_C(q_S + 1)}\varepsilon.$$

This ends the proof □

出席國際會議心得報告

申請人姓名：左瑞麟

服務單位及職稱：國立政治大學 資訊科學系 助理教授

會議名稱：

（英文）The 7th International Conference on Cryptology and Network Security (CANS2008)

（中文）第七屆密碼學與網路安全國際會議

會議地點：香港, **HKU Town Centre, Admiralty**

會議時間：2008 年 12 月 4 日至 12 月 6 日

論文題目：Efficient and short certificateless signature

參加會議經過：

CANS2008 提供全球密碼學與網路安全的專家學者一個研究交流的平台，持續推動資訊安全領域的發展。今年是第七屆，由香港城市大學所主辦，舉行為期二天之會議。會中邀請全球知名之專家學者演講。第一場邀請演講為 AT&T 實驗室的 Juan Garay 博士。 演講題目是”Sound and Fine-grain Specification of Cryptographic Tasks”。介紹密碼學的安全性證明，尤其在 Universal Composability(UC)安全的部

分。第二場邀請演講為北京清華大學的王小雲教授。演講題目為 Cryptanalysis on MACs。王教授是全球最著名的 hash function 專家。在 2005 年發現了 MD5 及 SHA-1 的 collision。這一次他們針對 HMAC/NMAC-MD5 找到了一些弱點攻擊。舉例來說，他們可以在經過 2^{97} 次的 query 之後，找到 128bit 的 MD5-MAC 的 subkey。

與會心得

有幸參與 CANS2008 年會，與來自世界各地的專家學者齊聚一堂， 針對密碼與資訊安全的相關議題相互討論，彼此交流，分享成果及實務經驗，實在是獲益良多。此次申請者的研究論文是關 於免憑證簽名方面，提供了一個安全有效的免憑證短簽名方案。我們在演講完畢之後，獲得了許多迴響及建議。未來將針對這些建議，繼續改良我們的方案。

攜回資料

The proceedings of the 7th International Conference on Cryptology and Network Security (CANS2008 會議論文摘要)

Efficient and Short Certificateless Signature

Raylin Tso[1,⋆], Xun Yi[2], and Xinyi Huang[3,⋆⋆]

[1] Department of Computer Science, National Chengchi University, Taiwan
raylin@cs.nccu.edu.tw

[2] School of Computer Science and Mathematics, Victoria University, Australia
Xun.Yi@vu.edu.au

[3] Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
xh068@uow.edu.au

Abstract. A certificateless signature (CLS) scheme with short signature size is proposed in this paper. Our scheme is as efficient as BLS short signature scheme in both communication and computation, and therefore turns out to be more efficient than other CLS schemes proposed so far. We provide a rigorous security proof of our scheme in the random oracle model. The security of our scheme is based on the k-CAA hard problem and a new discovered hard problem, namely, modified k-CAA problem. Our scheme can be applied to systems where signatures are typed in by human or systems with low-bandwidth channels and/or low-computation power, such as PDAs or cell phones.

Keywords: Bilinear pairing, certificateless signature, random oracle, short signature.

1 Introduction

Nowadays, the main difficulty in developing secure systems based on public key cryptography is the deployment and management of infrastructures to support the authenticity of cryptographic keys. The general approach to solve this problem is to use a Public Key Infrastructure (PKI) in which a trusted authority, called Certification Authority (CA), issues certificates to bind users and their public keys. However, the PKI is costly to use as it involves certificate revocation, storage, distribution, and verification.

In order to overcome the above mentioned problem, identity-based (ID-based) cryptography was firstly introduced by Shamir [19] in 1984. In an ID-based cryptosystem, one can use its unique identifier (e.g., names or e-mail addresses) as the public key. The user's identifier is publicly known and thus does not need certificates to prove its authenticity. Consequently, the problems associated with

[⋆] Supported by National Science Council of Taiwan (NSC 97-2218-E-004-002).

[⋆⋆] Supported by the National Natural Science Foundation of China (No. 60673070) and the Natural Science Foundation of Jiangsu Province (No. BK2006217).

M.K. Franklin, L.C.K. Hui, and D.S. Wong (Eds.): CANS 2008, LNCS 5339, pp. 64–79, 2008.

certificates can be eliminated. However, ID-based cryptosystems have an inherent key escrow issue as a third party "Private Key Generator" (PKG) generates the private keys for all users in the system. Therefore, the PKG must be fully trusted in ID-based cryptosystems.

Certificateless cryptography, firstly introduced by Al-Riyami and Paterson [2] in 2003, intends to solve the key escrow issue inherent in ID-based cryptography, and meanwhile to eliminate the use of certificates as in the conventional PKI. In a certificateless cryptosystem, private keys of users are generated by not only the PKG but also users themselves. In other words, PKG only issues a partial private key to each user while the user independently generates its additional public/secret key pair. Consequently, the PKG is unable to obtain secret keys of users. The cryptographic operations in certificateless system can be performed successfully only when both the partial private key and the secret key are known. In this way, the key escrow problem can be overcome. Following Al-Riyami and Paterson's pioneering work [2], many certificateless schemes have been proposed in recent years, such as [1,10,12,15,16,21,22,23,25] and etc..

1.1 Motivations

In the definition of the security model for certificateless signature (CLS) schemes, some papers (e.g., [1,14,15,17]) assume that the adversary should be allowed to obtain signatures signed with false public keys chosen by the adversary. But, in real world, the signatures that a "realistic" adversary can obtain are generated by a signer using the partial private key and the secret key corresponding to its original public key. Therefore, the adversary defined in those security models seems to enjoy more power than it could have in the real world. This assumption provides a higher security for the schemes on one hand but also limits the efficiency of the schemes on the other hand. This is because CLS schemes with a high security level usually sacrifice some efficiency in computation and/or communication and may not be practical for systems with low-bandwidth channels and/or low-computation power, such as PDAs or cell phones.

Except for the scheme proposed by Huang *et al.* [15], no secure CLS scheme has a short size of signature, although many short signatures in traditional PKI have been proposed [8,9,24]. As mentioned in [6], there are several important practical reasons for the desirableness of short signatures. For example, battery life is the major limitation on wireless devices such as PDAs, cell phones, RFID chips and sensors. Communicating even one bit of data uses significantly more power than executing one 32-bit instruction [3]. Reducing the number of bits to communicate saves power and is important to increase battery life. Also, in many settings, communication is not reliable, and thus the number of bits one has to communicate should be kept as few as possible. This inspired us to propose a more efficient certificateless short signature scheme.

1.2 Our Contributions

In this paper, on the basis of BLS short signature scheme [9], an efficient certificateless signature scheme with short signature size is proposed. Our scheme is as efficient as BLS short signature scheme (which is the traditional PKI model) in both communication and computation, and turns out to be more efficient than other CLS schemes proposed so far. This is achieved at the cost of stronger complexity assumptions.

In addition, as mentioned in [15], the security model defined in some CLS schemes (e.g., [16,21]) assume that, when an adversary queries the oracle **Public-Key-Replace** to replace a real public key with a false public key chosen by itself, the adversary is required to provide both the false public key and the corresponding secret value as the input. This is unreasonable since an adversary may pick a random public key for which the corresponding secret value is unknown even for himself. In other words, this definition may not cover the case in which an adversary may successfully forge a new signature with a false public key without knowing the corresponding secret value (to the false public key). Our definition for CLS scheme does not have such a problem and an adversary is not required to provide a secret value corresponding to a false public key as the input to the oracle **Public-Key-Replace**.

Based on the k-CAA problem, we define a new hard problem named "modified k-CAA problem". Assuming the hardness of these problems, we provide a rigorous security proof for our scheme in the random orale model .

The rest of this paper is organized as follows. In Section 2, we give some preliminaries (including the new discovered hardness assumption) which will be required throughout this paper. Section 3 is the presentation of our certificateless short signature scheme and in Section 4, we give the security proofs for our new scheme. Section 5 gives the performance comparison of our scheme with other schemes and the conclusion is given in Section 6.

2 Preliminaries

Before presenting our results, we first briefly review the notion of certificateless signature and its security definition. We will also review the definition for groups equipped with a bilinear map, and precisely state the hardness assumptions.

2.1 Certificateless Signatures

Following the definition in [2], a certificateless signature scheme is specified by seven randomized algorithms: **Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Sign** and **Verify**.

Setup. This algorithm takes as input a security parameter 1^k and returns the system parameters *params* and the master secret key *msk*. Usually, this algorithm is run by the KGC. We assume throughout that *params* are publicly and authentically available, but that only the KGC knows *msk*.

Partial-Private-Key-Extract. This algorithm takes the system parameter *params*, the master secret key *msk* and an identity *ID* as input. It returns a partial private key D_{ID}. Usually, this algorithm is run by the KGC and its output is transported to the identity *ID* over a confidential and authentic channel.

Set-Secret-Value. This algorithm takes as input the system parameter *params* and an identity *ID* as input and outputs a secret value x_{ID}. This algorithm is run by the identity *ID* for itself.

Set-Private-Key. This algorithm takes the system parameter *params*, a partial private key D_{ID} and a secret value x_{ID} of an identity *ID* as input. The value x_{ID} is used to transform D_{ID} into the (full) private key SK_{ID}. The algorithm returns SK_{ID}. This algorithm is run by the identity *ID* for itself.

Set-Public-Key. This algorithm takes the system parameter *params*, an identity *ID* and the identity's private key PK_{ID} as input. It outputs the public key PK_{ID} for the identity *ID*.

Sign. This algorithm takes the system parameter *params*, an identity *ID*, the private key SK_{ID} of *ID* and a message M as input. It outputs a certificateless signature σ.

Verify. This algorithm takes the system parameter *params*, an identity *ID*, the identity's public key PK_{ID} and a message/signature pair (M, σ) as input. It output *true* if the signature is correct, or $false$ otherwise.

2.2 Security Model

In this section, we discuss the definition of the security for a certificateless signature scheme.

For certificateless cryptosystems, the widely accepted notion of security was defined by Al-Riyami and Paterson in [2]. According to their definitions as well as the definitions in [25], there are two types of adversary with different capabilities:
Type I Adversary: This type of adversary $\mathcal{A}_I$ models a dishonest user who does not have access to the master key *msk* but has the ability to replace the public key of any entity with a value of his choice.
Type II Adversary: This type of adversary $\mathcal{A}_{II}$ models a malicious KGC who has access to the master key *msk* but *cannot perform public keys replacement*[1].

Generally, there are five oracles which can be accessed by the adversaries according to the game specifications which will be given later.

1. **Create-User:** On input an identity $ID \in \{0, 1\}^*$, if *ID* has already been created, nothing is to be carried out. Otherwise, the oracle runs the algorithms **Private-Key-Extract**, **Set-Secret-Value**, **Set-Public-Key** to obtain the partial private key D_{ID}, secret value x_{ID} and public key PK_{ID}. In this case, *ID* is said to be created. In both cases, PK_{ID} is returned.
2. **Public-Key-Replace:**[2] On input an identity *ID* and a user public key PK'_{ID}, the original user public key of *ID* is replaced with PK'_{ID} if *ID* has been created. Otherwise, no action will be taken.

[1] It is important that in certificateless cryptosystems, KGC must be semi-trusted and cannot perform the public key replacement. This is because that any adversary who knows the master key can impersonate anyone if he is allowed to replace the public key of the entity.

[2] Different from the security model defined in [16,21], in this oracle, an adversary is not required to provide the secret value x'_{ID} which is used to generated the public key PK'_{ID}.

3. **Secret-Value-Extract:** On input an identity, it returns the corresponding user secret key x_{ID} if ID has been created. Otherwise, returns a symbol $\perp$. Note that x_{ID} is the secret value associated with the original public key PK_{ID}. This oracle does not output the secret value associated with the replaced public key PK'_{ID}.
4. **Partial-private-Key-Extract:** On input an identity ID, it returns the partial private key D_{ID} if ID has been created. Otherwise, returns a symbol $\perp$.
5. **Sign:** On input an identity ID and a message $m \in \{0,1\}^*$, the signing oracle proceeds in one of the both cases below.
   - If ID has not been created, returns $\perp$.
   - If ID has been created, returns a valid signature σ such that $true \leftarrow$ Veify(m, σ, ID, PK_{ID}). Here PK_{ID} is the public key returned from the oracle **Create-User**.

The standard notion of security for a signature scheme is called existential unforgeability against adaptive chosen message attack defined by Goldwasser, Micali and Revist [11]. To define the existential unforgeability of a certificateless signature against Type I adversary $\mathcal{A}_I$ and Type II adversary $\mathcal{A}_{II}$, we define two games, one for $\mathcal{A}_I$ and the other for $\mathcal{A}_{II}$.

Game 1: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_I$.

Setup. The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_I$ is given *params* but the master secret key *msk* is kept by the challenger.

Queries. $\mathcal{A}_I$ adaptively access all the oracles defined in Section 2.2 in a polynomial number of times.

Forgery. Eventually, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if the following conditions hold true:

1. $true \leftarrow$ Verify$(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been submitted to the oracle **Sign**.
3. ID^* has never been submitted to the oracle **Partial-Private-Key-Extract** and **Secret-Value-Extract**.

Definition 1. Define $Adv_{\mathcal{A}_I}$ to be the probability that a Type I adaptively chosen message and chosen identity adversary $\mathcal{A}_I$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_I$ and the challenger. We say a certificateless signature scheme is secure against Type I attack, if, for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_I$, the success probability $Adv_{\mathcal{A}_I}$ is negligible.

Game 2: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_{II}$.

Setup. The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_{II}$ is given both *params* and *msk*.

Queries. $\mathcal{A}_{II}$ adaptively access all the oracles defined in Section 2.2 in a polynomial number of times.

Forgery. Eventually, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if the following conditions hold true:

1. $true \leftarrow \text{Verify}(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been queried to the oracle **Sign**.
3. ID^* has never been submitted to the oracle **Secret-Value-Extract**.

Definition 2. Define $Adv_{\mathcal{A}_{II}}$ to be the probability that a Type II adaptively chosen message and chosen identity adversary $\mathcal{A}_{II}$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_{II}$ and the challenger. We say a certificateless signature scheme is secure against Type II attack, if, for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{II}$, the success probability $Adv_{\mathcal{A}_{II}}$ is negligible.

Definition 3. A certificateless signature scheme is existentially unforgeable against adaptive chosen message and chosen identity attack if it is secure against both Type I and Type II attacks defined above.

2.3 Bilinear Groups and Complexity Assumptions

Let $\mathbb{G}_1, \mathbb{G}_2$ be two multiplicative cyclic groups of order p for some large prime p. Our scheme makes use of the *bilinear* map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between these two groups. The bilinear map should be satisfied with the following properties:

1. **Bilinear:** A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is bilinear if $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$ for all $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$.
2. **Non-degenerate:** The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$. Observe that since $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order, this implies that if g is a generator of $\mathbb{G}_1$, then $\hat{e}(g, g)$ is a generator of $\mathbb{G}_2$.
3. **Computable:** There is an efficient algorithm to compute $\hat{e}(g, h)$ for any $g, h \in \mathbb{G}_1$.

A bilinear map satisfying the three properties above is said to be an *admissible* bilinear map. We can make this map using the Weil pairing or the Tate pairing [4,5,9]

Next, we describe the complexity assumptions which are required for the security proof of our scheme.

We first introduce a problem given by Mitsunari *et al.* [18] which is called k-CAA (Collusion Attack Algorithm with k traitors) problem and then give a modified problem.

Definition 4. **k-CAA Problem** [18]
For $x, h_1, \cdots, h_k \in \mathbb{Z}_p^*$, and a generator g of $\mathbb{G}_1$. Given g, g^x and k pairs $(h_1, g^{(x+h_1)^{-1}}), \cdots, (h_k, g^{(x+h_k)^{-1}})$, output a new pair $(h^*, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$.

The k-CAA problem is believed to be hard. Mitsunari *et al.* firstly introduced this problem and gave a traitor tracing scheme [18] based on this problem. Although their application to tracing traitors is proved by Tô *et al.* [20] to be insecure, the k-CCA problem still remains to be hard without broken. Zhang et al. [24] recently gave a secure and efficient signature scheme based on the same problem.

In addition to the k-CAA problem, the security of our scheme also bases on a modified version of the original k-CAA problem. We call it as the Modified k-CAA Problem which is defined as follows:

Definition 5. Modified k-CAA Problem
For randomly picked $x, a, b, h_1, \cdots, h_k \in \mathbb{Z}_p^*$, and a generator g of $\mathbb{G}_1$. Let $g_1 = g^{ab} \neq g$. Given g, g^x, g^a, g^b, g^{bx} and k pairs $(h_1, g_1^{(x+h_1)^{-1}}), \cdots, (h_k, g_1^{(x+h_k)^{-1}})$, output either a new pair $(h^*, g_1^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$ or g_1.

Note that in the above definition, g_1 is not given to the problem. If we define $g_1 = g$ in the input, then g^a, g^b and g^{bx} are useless and can be ignored. In this case, the problem is to find a new pair $(h^*, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$.

3 The Proposed Certificateless Short Signature Scheme

In this section, we will describe our certificateless short signature scheme. It consists of the following algorithms:

Setup: Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups of some prime order $p \geq 2^k$, k be the security parameter of the scheme. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is an admissible bilinear pairing. Let $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be two secure cryptographic hash functions. KGC chooses a random number $s \in \mathbb{Z}_p^*$ and an arbitrary generator $g \in \mathbb{G}_1$. It sets $P_{pub} = g^s$, publishes $params = \{\mathbb{G}_1, \mathbb{G}_2, g, \hat{e}, H_0, H_1, P_{pub}\}$ and keeps the master secret key $msk = s$ secretly.

Partial-Private-Key-Extract: Given an entity's identity $ID \in \{0, 1\}^*$, KGC sets $Q_{ID} = H_0(ID)$ and computes the entity's partial private key $D_{ID} = Q_{ID}^s$. KGC transmits D_{ID} to ID over a confidential and authentic channel.

Set-Secret-Value: The entity ID chooses a random number $x_{ID} \in \mathbb{Z}_p^*$.

Set-Private-Key: The entity ID sets his private key as $SK_{ID} = (D_{ID}, x_{ID})$.

Set-Public-Key: Given x_{ID}, the entity ID computes the public key $PK_{ID} = (PK_1, PK_2) = (g^{x_{ID}}, Q_{ID}^{x_{ID}})$.

Sign: To sign a message $m \in \{0,1\}^*$, the entity ID first sets $h = H_1(m||ID||PK_{ID})$ and then computes the signature $\sigma = D_{ID}^{(x_{ID}+h)^{-1}}$.

Verify: Given a pair (m, σ) and ID's public key $PK_{ID} = (g^{x_{ID}}, Q_{ID}^{x_{ID}})$, any verifier first checks the equation $\hat{e}(PK_1, Q_{ID}) = \hat{e}(PK_2, g)$. If it holds, then computes $h = H_1(m||ID||PK_{ID})$ and checks the equation

$$\hat{e}(\sigma, PK_1 \cdot g^h) \stackrel{?}{=} \hat{e}(H_0(ID), P_{pub}).$$

If the equality holds, outputs *true*, otherwise, outputs *false*.

Correctness: If σ is a valid signature on m, then the correctness holds since

$$\begin{aligned}
&\hat{e}(\sigma, PK_1 \cdot g^h)\\
&= \hat{e}(D_{ID}^{(x_{ID}+h)^{-1}}, g^{x_{ID}} \cdot g^h) = \hat{e}(H_0(ID)^{s(x_{ID}+h)^{-1}}, g^{x_{ID}+h})\\
&= \hat{e}(H_0(ID), g)^{s(x_{ID}+h)^{-1}(x_{ID}+h)} = \hat{e}(H_0(ID), g)^s\\
&= \hat{e}(H_0(ID), g^s) = \hat{e}(H_0(ID), P_{pub}).
\end{aligned}$$

4 Security Proofs

Theorem 1. Unforgeability against Type I Adversary: If there exists a Type I adaptively chosen message and chosen *ID* adversary $\mathcal{A}_I$ who can ask at most q_C **Create-User** queries, q_{KEx} **Partial-Private-Key-Extract** queries, q_{VEx} **Secret-Value-Extract** queries and q_S **sign** queries, respectively, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, using $\mathcal{A}_I$ as a black box, can solve the modified k-CAA problem [Definition 5] (where $k \geq q_S$ and is in proportion to the number of the H_1-hash queries) with probability $Adv_{\mathcal{F}}^{mk-CAA} \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1 - \frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon$.

Proof: If there exists an adversary $\mathcal{A}_I$ who can break the unforgeability of the proposed scheme via Type I attack, then, we can construct another adversary $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_I$ as a black-box and solve the modified k-CAA problem.

Let g be a generator of $\mathbb{G}_1$, x, a, b be three random numbers of $\mathbb{Z}_p^*$ and $g_1 = g^{ab} \in \mathbb{G}_1$. Let $h_1, \cdots, h_k \in \mathbb{Z}_p^*$ be k random numbers. $\mathcal{F}$ is given the challenge $\{g, g^x, g^a, g^b, g^{bx}, (h_1, g_1^{(x+h_1)^{-1}}), \cdots, (h_k, g_1^{(x+h_k)^{-1}})\}$. The purpose of $\mathcal{F}$ is either to find a new pair $(h^*, g_1^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$ or to find g_1, which are the solutions to the modified k-CAA problem.

Setup: In order to solve the problem, $\mathcal{F}$ utilizes $\mathcal{A}_I$ as a black-box. To get the black-box $\mathcal{A}_I$ run properly, $\mathcal{F}$ will simulate the environments of the proposed scheme and the oracles which $\mathcal{A}_I$ can access. In this proof, we regard the hash functions H_0, H_1 as random oracles. $\mathcal{F}$ starts by picking an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, and sets $P_{pub} = g^a$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ to $\mathcal{A}_I$ and allows $\mathcal{A}_I$ to run.

Due to the ideal randomness of the H_1-hash, we may assume that $\mathcal{A}_I$ is well-behaved in the sense that it always requests a H_1-hash of $m||ID||PK_{ID}$ before it requests a signature for m signed by ID's public key PK_{ID}. In addition, it always requests a H_1-hash of $m^*||ID^*||PK_{ID^*}$ that it outputs as its forgery. It is trivial to modify any adversary-algorithm $\mathcal{A}_I$ to have this property.

Query: At any time, $\mathcal{A}_I$ is allowed to access the following oracles in a polynomial number of times. These oracles are all simulated by $\mathcal{F}$.

1. **Create-User:** $\mathcal{A}_I$ can query this oracle by given an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \cdots, q_C\}$.

(1) If $i \neq t$, $\mathcal{F}$ chooses $d_i, x_i \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_i) = g^{d_i}$, $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)}) = (g^{x_i}, g^{d_i x_i})$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = H_0(ID_i)^a = g^{ad_i} = P_{pub}^{d_i}$ and the secret value is $x_{ID_i} = x_i$.

(2) If $i = t$, $\mathcal{F}$ sets $H_0(ID_t) = g^b$ and $PK_{ID_t} = (PK_{(ID_t,1)}, PK_{(ID_t,2)}) = (g^x, g^{bx})$. In this case, $\mathcal{F}$ will set $D_{ID_t} = x_{ID_t} = \perp$ which means that it cannot compute the secret value and the partial private key of ID_t.

In both cases, returns $H_0(ID_i)$ and PK_{ID_i}.

2. **Partial-Private-Key-Extract:** At any time, $\mathcal{A}_I$ can query the oracle by given an identity ID_i. $\mathcal{F}$ outputs a symbol $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns $D_{ID_i} = g^{ad_i}$. Otherwise, $\mathcal{F}$ returns *failure* and terminates the simulation.
3. **Public-Key-Replace:** $\mathcal{A}_I$ can request to replace public key PK_{ID_i} of an entity ID_i with new public key PK'_{ID_i} chosen by $\mathcal{A}_I$ itself. $\mathcal{F}$ replaces the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, outputs $\perp$. Here, to replace a public key, the secret value corresponding to the new public key is not required.
4. **Secret-Value-Extract:** Given ID_i chosen by $\mathcal{A}_I$, outputs $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns x_{ID_i} to $\mathcal{A}_I$. Otherwise, $i = t$ and $\mathcal{F}$ reports *failure* and terminates the simulation.
5. H_1 **Queries:** $\mathcal{A}_I$ can query the random oracle H_1 at any time on an input $\omega_i = (m_l||ID_j||PK_{ID_k})$. For i-th H_1 query asked by $\mathcal{A}_I$ on input ω_i, $\mathcal{F}$ first checks if $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$ or not. Here PK_{ID_t} is the original public key.
   - If $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$, then $\mathcal{F}$ first flips a biased coin which outputs a value $c_i = 1$ with probability ζ, and $c_i = 0$ with probability $1 - \zeta$ (the value of ζ will be optimized later).

     (1) If $c_i = 1$, $\mathcal{F}$ picks a random value $h'_i \in \mathbb{Z}_p^*$ where $h'_i \notin \{h_1, \cdots, h_k\}$ and responds h'_i to $\mathcal{A}_I$ as the value of $H_1(\omega_i)$.

     (2) If $c_i = 0$, $\mathcal{F}$ returns a value $h''_i \in_R \{h_1, \cdots, h_k\}$ as the output of $H_1(\omega_i)$ where h''_i must be a fresh value which means that it has not been assigned as an output of H_1 queries before.
   - Otherwise, $\mathcal{F}$ picks and responds with a random value $\mu_i \in \mathbb{Z}_p^*$.

   In either cases, $\mathcal{F}$ records (ω_i, h'_i, c_i), (ω_i, h''_i, c_i) or (ω_i, μ_i) to a H_1-*List* which is initially empty.
6. **Sign:** For each sign query on an input (m_l, ID_j), output $\perp$ if ID_j has not been created. For any input (m_l, ID_j) with ID_j which has already been created, since we assume that $\mathcal{A}_I$ is well-behaved, we know that $\mathcal{A}_I$ has already queried the random oracle H_1 on the input $\omega_i = (m_l||ID_j||PK_{ID_j})$.
   - If $ID_j \neq ID_t$, $\mathcal{F}$ uses the private key (x_{ID_j}, D_{ID_j}) of ID_j and $\mu_i = H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_l and the identity ID_j.
   - If $ID_i = ID_t$, then, $\mathcal{F}$ first checks the H_1-*List*.

     (1) If $c_i = 1$, $\mathcal{F}$ reports *failure* and terminates the simulation.

     (2) Otherwise, $c_i = 0$ and $h''_i = H_1(m_l||ID_t||PK_{ID_t})$ is on the H_1-*List*. For easy of description, we assume $h''_i = h_i \in \{h_1, \cdots, h_k\}$.

$\mathcal{F}$ then returns $\sigma_i = g_1^{(x+h_i)^{-1}}$. Note that

$$\hat{e}(\sigma_i, PK_{(ID_t,1)} \cdot g^{h_i}) = \hat{e}(g_1^{(x+h_i)^{-1}}, g^x \cdot g^{h_i}) = \hat{e}(g_1, g)$$
$$= \hat{e}(g^{ab}, g) = \hat{e}(g^b, g^a) = \hat{e}(H_0(ID_t), P_{pub}).$$

Therefore, σ_i is a valid signature on m_l and ID_t.

Forgery: After all the queries, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK_{ID^*} = (PK_{(ID^*,1)}, PK_{(ID^*,2)}), m^*, \sigma^*)$ and wins the game.

If σ^* is a valid forgery, then $h^* = H_1(m^*||ID^*||PK_{ID^*})$ which is on the H_1-*List*, and $\hat{e}(\sigma^*, PK_{(ID^*,1)} \cdot g^{h^*}) = \hat{e}(H_0(ID^*), P_{pub})$ where $PK_{ID^*} = g^{x^*}$ may be a new public key replaced by $\mathcal{A}_I$ or the original public key generated by the oracle **Create-User**. In addition, $\hat{e}(PK_{(ID^*,1)}, Q_{ID^*}) = \hat{e}(PK_{(ID^*,2)}, g)$ if $\mathcal{A}_I$ wins the game. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs $failure$ and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.

(1) If $c^* = 0$, $\mathcal{F}$ outputs $failure$ and terminates the simulation.

(2) Otherwise, $c^* = 1$ and $h^* \notin \{h_1, \cdots, h_k\}$. If $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = (PK_{(ID_t,1)}, PK_{(ID_t,2)})$ is the original public key generated by the oracle, then, $\mathcal{F}$ outputs a new pair $(h^*, \sigma^*) = (h^*, g_1^{(x+h^*)^{-1}})$ which will be the solution to the modified k-CAA problem. If $(PK_{(ID^*,1)}, PK_{(ID^*,2)})$ is a new public key replaced by $\mathcal{A}_I$, then, using the knowledge of exponent assumption introduced in [7,13], $\mathcal{F}$ can either extract x^* if $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = (g^*, g^{bx^*})$ is generated from (g, g^b) or extract r if $(PK_{(ID^*,1)}, PK_{(ID^*,2)}) = ((g^x)^r, (g^{bx})^r)$ is generated from (g^x, g^{bx}). Consequently, $g_1 = (\sigma^*)^{(x^*+h^*)}$ can be computed if x^* extracted or a new pair $(h', g_1^{(x+h')^{-1}}) = (h^*/r, (\sigma^*)^r)$ can be found if r extracted, which is also the solution to the modified K-CAA problem.

It remains to compute the probability that $\mathcal{F}$ solves the modified k-CAA problem. Actually, $\mathcal{F}$ succeeds if:

Λ_1 : $\mathcal{F}$ does not abort during the simulation.
Λ_1 : σ^* is a valid forgery on (ID^*, PK_{ID^*}, m^*).
Λ_1 : $ID^* = ID_t$ and $c^* = 1$.

The advantage of $\mathcal{F}$ is $Adv_{\mathcal{F}}^{BCk-CAA} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2]$. If Λ_1 happens, then:

- $\mathcal{F}$ does not output $failure$ during the simulation of the oracle **Partial-Private-Key-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{PKEx}}$.
- $\mathcal{F}$ does not output $failure$ during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{VEx}}$.
- $\mathcal{F}$ does not output $failure$ during the simulation of **sign** oracle. This happens with probability $(1 - \frac{1}{q_C}\zeta)^{q_S} \geq (1 - \zeta)^{q_S}$.

Consequently, $Pr[\Lambda_1] \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1 - \zeta)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \frac{\zeta}{q_C}$. Therefore, $Adv_{\mathcal{F}}^{BCk-CAA} \geq (1 - \frac{1}{q_C})^{q_{PKEx}+q_{VEx}}$

Table 1. Performance Evaluation

	Type	PK-Size (bits)	Sig-Length (bits)	Sign	Verify
Ours	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$1\hat{e} + 1E_{G_1}$
BLS [9]	No	$\|G_1\|$ (≈ 160)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$1\hat{e} + 1E_{G_1}$
Scheme I [15]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\|$ (≈ 160)	$1E_{G_1}$	$2\hat{e}$
ZWXF [25]	CLS	$\|G_1\|$ (≈ 160)	$2\|G_1\|$ (≈ 320)	$3E_{G_1}$	$3\hat{e}$
HSMZ [16]	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\| + 1\|q\|$ (≈ 320)	$2E_{G_1} + 1E_{G_2}$	$1\hat{e} + 1E_{G_2}$
AP03 [2]	CLS	$2\|G_1\|$ (≈ 320)	$\|G_1\| + 1\|q\|$ (≈ 320)	$2E_{G_1} + 1E_{G_2}$	$1\hat{e} + 1E_{G_2}$
eCLS [10]	CLS	$\|G_1\|$ (≈ 160)	$2\|G_1\|$ (≈ 320)	$2E_{G_1}$	$2\hat{e} + 1E_{G_1}$
Scheme II [15]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\| + 2\|p\|$ (≈ 480)	$3E_{G_1} + 1E_{G_2}$	$1\hat{e} + 3E_{G_1} + 1E_{G_2}$
oCLS [10]	CLS	$\|G_1\|$ (≈ 160)	$\|G_1\| + \|G_2\|$ (≈ 1184)	$2E_{G_1}$	$1\hat{e} + 1E_{G_2}$

$(1-\zeta)^{q_S}\frac{\zeta}{q_C}\varepsilon$. The function $\zeta(1-\zeta)^{q_S}$ is maximized at $\zeta = \frac{1}{q_S+1}$. Therefore,

$$Adv_{\mathcal{F}}^{BCk-CAA} \geq (1-\frac{1}{q_C})^{q_{PKEx}+q_{VEx}}(1-\frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon.$$

This ends the proof. □

Theorem 2. Unforgeability against Type II Adversary: If there exists a Type II adaptively chosen message and chosen *ID* adversary $\mathcal{A}_{II}$ who can ask at most q_C **Create-User** queries, q_{VEx} **Secret-Value-Extract** queries and q_S **Sign** queries, respectively, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, using $\mathcal{A}_{II}$ as a black box, can solve the k-CAA problem [Definition 4] (where $k \geq q_S$ and is in proportion to the number of the H_1-hash queries) with probability $Adv_{\mathcal{F}}^{k_CAA} \geq (1-\frac{1}{q_C})^{q_{VEx}}(1-\frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon$.

Proof: The proof is similar to that of proving Theorem 1 with a little modification. See Appendix for details. □

Theorem 1 is proved in a relatively weaker model than the normal one. That is, we do not allow the adversary to obtain valid signatures according to the replaced public key.

As mentioned in Section 1, this model is also acceptable as the signatures that a "realistic" adversary can obtain are usually generated by a signer under its original public key. Therefore, this modification is reasonable and Huang et al.'s first scheme with short signature size [15] is also analyzed in this weak model.

5 Performance Comparison

In this section, we compare our certificateless short signature scheme with other existing CLS schemes and BLS short signature scheme [9] from the aspect of communication cost and computation cost in signature signing and verification, respectively.

In the comparison, the operations such as $\hat{e}(g, g)$, $\hat{e}(PK_1, Q_{ID}) = \hat{e}(PK_2, g)$ or $\hat{e}(H_0(ID), P_{pub})$ are pre-computable or only need to be computed once. Therefore, these computations are neglected in the comparison. In Table 1, certificateless signature schemes are marked with "CLS". Other schemes are marked with"No". We denote by $\hat{e}$ a computation of the pairing, E_{G_1} an exponentiation in $\mathbb{G}_1$, and E_{G_2} an exponentiation in $\mathbb{G}_2$. Usually, pairing operations cost much more than other computations. One $\hat{e}$ operation is about 10 times more expensive than one $E_{(.)}$ operation.

We can see in Table 1 that our scheme is as efficient as BLS short signature [9] but our scheme is certificateless whereas BLS scheme is not. This means there is no need to verify a certificate in our scheme while using BLS scheme, a verifier needs to verify the certificate in order to confirm the correctness of the public key, as in the conventional Public key Infrastructure (PKI), which is generally considered to be costly to use and manage. From this point of view, our scheme is superior than BLS short signature scheme.

Among all certificateless signature schemes, Huang et al.'s first scheme in [15] is the only signature scheme providing short signature-length (about 160 bits) as ours. However, our scheme is more efficient than their scheme in the verification phase. To the best of our knowledge, our scheme is the most efficient CLS scheme in the aspects of both communication and computation costs.

6 Conclusion

In this paper, we proposed a certificateless signature scheme which is as efficient as BLS short signature. We also defined a new hard problem "modified k-CAA problem" based on the k-CAA problem. The security of the proposed scheme is proved in the random oracle model under the hardness of k-CAA problem and modified k-CAA problem.

References

1. Au, M.H., Chen, J., Liu, J.K., Mu, Y., Wong, D.S., Yang, G.: Malicious KGC attacks in certificateless cryptography. In: Proceedings of ASIACCS 2007, pp. 302–311 (2007)
2. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
3. Barr, K., Asanovic, K.: Energy aware lossless data compression. In: Proceedings of the ACM Conference on Mobile Systems, Applications, and Services (MobiSys) (2003)
4. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithm for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–369. Springer, Heidelberg (2002)
5. Barreto, P.S.L.M., Lynn, B., Scott, M.: On the selection of pairing-friendly groups. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 17–25. Springer, Heidelberg (2004)

6. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM Confetence on Computer and Communication Security, pp. 390–398 (2006)
7. Bellare, M., Palacio, A.: The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)
8. Boneh, D., Boyen, X.: Short signatures withou rando oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
9. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–533. Springer, Heidelberg (2001)
10. Choi, K.Y., Park, J.H., Hwang, J.Y., Lee, D.H.: Efficient certificateless signature schemes. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 443–458. Springer, Heidelberg (2007)
11. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal of Computing 17(2), 281–308 (1988)
12. Gorantla, M.C., Saxena, A.: An efficient certificateless signature scheme. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS, vol. 3802(II), pp. 110–116. Springer, Heidelberg (2005)
13. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 408–423. Springer, Heidelberg (1998)
14. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Certificatelss signature: a new security model and an improved generic construction. Designs, Codes and Cryptography 42(2), 109–126 (2007)
15. Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless signature revisted. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 308–322. Springer, Heidelberg (2007)
16. Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the security of certificateless signature schemes from Asiacrypt 2003. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (eds.) CANS 2005. LNCS, vol. 3810, pp. 13–25. Springer, Heidelberg (2005)
17. Liu, J.K., Au, M.H., Susilo, W.: Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: Proceedings of ASIACCS 2007, pp. 273–283 (2007)
18. Mitsunari, S., Sakai, R., Kasahara, M.: A new traitor tracing. Journal of IEICE Trans. Fundamentals E85-A(2), 481–484 (2002)
19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
20. Tô, V., Safavi-Naini, R., Zhang, F.: New traitor tracing schemes using bilinear map. In: Proceedings of 2003 DRM Workshop, pp. 67–76 (2003)
21. Yap, W.L., Heng, S.H., Goi, B.M.: An efficient certificteless signature. In: Zhou, X., Sokolsky, O., Yan, L., Jung, E.-S., Shao, Z., Mu, Y., Lee, D.C., Kim, D.Y., Jeong, Y.-S., Xu, C.-Z. (eds.) EUC Workshops 2006. LNCS, vol. 4097, pp. 322–331. Springer, Heidelberg (2006)
22. Yap, W.L., Chow, S.S.M., Heng, S.H., Goi, B.M.: Security Mediated Certificateless Signatures. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 459–477. Springer, Heidelberg (2007)

23. Yum, D.H., Lee, P.J.: Generic construction of certificateless signature. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 200–211. Springer, Heidelberg (2004)
24. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from binilear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004)
25. Zhang, Z., Wong, D.S., Xu, J., Feng, D.: Certificateless public-key signature: security model and efficiet construction. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 293–308. Springer, Heidelberg (2006)

Appendix

Proof of Theorem 2

Proof: If there exists an adversary $\mathcal{A}_{II}$ who can break the unforgeability of the proposed scheme via Type II attack, then, we can construct another adversary $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_{II}$ as a black-box and solve the k-CCA problem.

Let g be a generator of $\mathbb{G}_1$, and $x, h_1, \cdots, h_k \in \mathbb{Z}_p^*$ be $k+1$ random numbers. $\mathcal{F}$ is given the challenge $\{g, g^x, (h_1, g^{(x+h_1)^{-1}}), \cdots, (h_k, g^{(x+h_k)^{-1}})\}$. The purpose of $\mathcal{F}$ is to output a tuple $(h, g^{(x+h^*)^{-1}})$ for some $h^* \notin \{h_1, \cdots, h_k\}$, which is the solution to the k-CAA problem.

Setup: In order to solve the problem, $\mathcal{F}$ utilizes $\mathcal{A}_{II}$ as a black-box. To get the black-box $\mathcal{A}_{II}$ run properly, $\mathcal{F}$ will simulate the environments of the proposed scheme and the oracles which $\mathcal{A}_{II}$ can access. In this proof, we regard the hash functions H_0, H_1 as random oracles. $\mathcal{F}$ starts by picking an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, and sets $P_{pub} = g^s$, where s is randomly chosen from $\mathbb{Z}_p^*$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ together with the master secret key s to $\mathcal{A}_{II}$ and allows $\mathcal{A}_{II}$ to run.

Due to the ideal randomness of the H_1-hash, we may assume that $\mathcal{A}_{II}$ is well-behaved in the sense that it always requests a H_1-hash of $m||ID||PK_{ID}$ before it requests a signature for m signed by ID's public key PK_{ID}. In addition, it always requests a H_1-hash of $m^*||ID^*||PK_{ID^*}$ that it outputs as its forgery. It is trivial to modify any adversary-algorithm $\mathcal{A}_{II}$ to have this property.

Query: At any time, $\mathcal{A}_{II}$ is allowed to access the following oracles in a polynomial number of times. These oracles are all simulated by $\mathcal{F}$. Different from the proof for Type I adversary, there is no oracle **Partial-Private-Key-Extract**. This is because that $\mathcal{A}_{II}$ has already obtained the master secret key s so he can compute the partial private key (i.e., $D_{ID} = H_0(ID)^s$)) of any entity using the master key s.

1. **Create-User:** $\mathcal{A}_{II}$ can query this oracle by given an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \cdots, q_C\}$.
   (1) If $i \neq t$, $\mathcal{F}$ chooses $d_i, x_i \in_R \mathbb{Z}_p^*$ and computes $H_0(ID_i) = g^{d_i}$, $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)}) = (g^{x_i}, g^{x_i d_i})$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = g^{sd_i}$ and the secret value is $x_{ID_i} = x_i$.

(2) If $i = t$, $\mathcal{F}$ chooses $d_t \in_R \mathbb{Z}_p^*$ and computes $H_0(ID_t) = g^{d_t}$. However, $\mathcal{F}$ sets $PK_{ID_t} = (PK_{(ID_t,1)}, PK_{(ID_t,2)}) = (g^x, g^{xd_t})$. In this case, $\mathcal{F}$ will set $D_{ID_t} = g^{sd_t}$ and $x_{ID_t} = \perp$ which means that it cannot compute the secret value of ID_t.

In both cases, returns $H_0(ID_i)$ and PK_{ID_i}.

2. **Public-Key-Replace:** $\mathcal{A}_{II}$ can request to replace public key PK_{ID_i} of an entity ID_i with new public key PK'_{ID_i} chosen by $\mathcal{A}_{II}$ itself. $\mathcal{F}$ replaces the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, outputs $\perp$. Here, to replace a public key, the secret value corresponding to the new public key is not required.
3. **Secret-Value-Extract:** Given ID_i chosen by $\mathcal{A}_{II}$, outputs $\perp$ if ID_i has not been created. If ID_i has been created and $i \neq t$, $\mathcal{F}$ returns x_{ID_i} to $\mathcal{A}_{II}$. Otherwise, $i = t$ and $\mathcal{F}$ reports $failure$ and terminates the simulation.
4. H_1 **queries:** $\mathcal{A}_{II}$ can query the random oracle H_1 at any time on an input $\omega_i = (m_l||ID_j||PK_{ID_k})$. For i-th H_1 query asked by $\mathcal{A}_{II}$ on input ω_i, $\mathcal{F}$ first checks if $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$ or not. Here PK_{ID_t} is the original public key.
   - If $ID_j = ID_t$ and $PK_{ID_k} = PK_{ID_t}$, then $\mathcal{F}$ first flips a biased coin which outputs a value $c_i = 1$ with probability ζ, and $c_i = 0$ with probability $1 - \zeta$ (the value of ζ will be optimized later).
     (1) If $c_i = 1$, $\mathcal{F}$ picks a random value $h'_i \in \mathbb{Z}_p^*$ where $h'_i \notin \{h_1, \cdots, h_k\}$ and responds h'_i to $\mathcal{A}_{II}$ as the value of $H_1(\omega_i)$.
     (2) If $c_i = 0$, $\mathcal{F}$ returns a value $h''_i \leftarrow_R \{h_1, \cdots, h_k\}$ as the output of $H_1(\omega_i)$ where h''_i must be a fresh value which means that it has not been assigned as an output of H_1 queries before.
   - Otherwise, $\mathcal{F}$ picks and responds with a random value $\mu_i \in \mathbb{Z}_p^*$.

   In either cases, $\mathcal{F}$ records (ω_i, h'_i, c_i), (ω_i, h''_i, c_i) or (ω_i, μ_i) to a H_1-*List* which is initially empty.
5. **Sign:** For each sign query on an input (m_l, ID_j), output $\perp$ if ID_j has not been created. For any input (m_l, ID_j) with ID_j which has already been created, since we assume that $\mathcal{A}_{II}$ is well-behaved, we know that $\mathcal{A}_{II}$ has already queried the random oracle H_1 on the input $\omega_i = (m_l||ID_j||PK_{ID_j})$.
   - If $ID_j \neq ID_t$, $\mathcal{F}$ uses the private key (x_{ID_j}, D_{ID_j}) of ID_j and $\mu_i = H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_l and the identity ID_j.
   - If $ID_i = ID_t$, then, $\mathcal{F}$ first checks the H_1-*List*.
     (1) If $c_i = 1$, $\mathcal{F}$ reports $failure$ and terminates the simulation.
     (2) Otherwise, $c_i = 0$ and $h''_i = H_1(m_l||ID_t||PK_{ID_t})$ is on the H_1-*List*. For easy of description, we assume $h''_i = h_i \in \{h_1, \cdots, h_k\}$. $\mathcal{F}$ then returns $\sigma_i = g^{sd_t(x+h_i)^{-1}}$. Note that

$$\hat{e}(\sigma_i, PK_{(ID_t,1)} \cdot g^{h_i}) = \hat{e}(g^{sd_t(x+h_i)^{-1}}, g^x \cdot g^{h_i}) = \hat{e}(g^{sd_t}, g)$$
$$= \hat{e}(g, g)^{sd_t} = \hat{e}(g^{d_t}, g^s) = \hat{e}(H_0(ID_t), P_{pub}).$$

Therefore, σ_i is a valid signature on m_l and ID_t.

Forgery: After all the queries, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK_{ID^*} = (PK_{(ID^*,1)}, PK_{(ID^*,2)}), m^*, \sigma^*)$ and wins the game.
If σ^* is a valid forgery, then $h^* = H_1(m^*||ID^*||PK_{ID^*})$ which is on the H_1-*List*, and

$$\hat{e}(\sigma^*, PK_{(ID^*,1)} \cdot g^{h^*}) = \hat{e}(H_0(ID^*), P_{pub})$$

where $PK_{(ID^*,1)} = g^{x_{ID^*}}$ must be the original public key generated by the oracle **Create-User**. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs $failure$ and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.

(1) If $c^* = 0$, $\mathcal{F}$ outputs $failure$ and terminates the simulation.
(2) Otherwise, $c^* = 1$ and $h^* \notin \{h_1 \cdots, h_k\}$. $\mathcal{F}$ computes $\xi = (\sigma^*)^{(sd_t)^{-1}}$ and outputs the tuple $(h^*, \xi) = (h^*, g^{(x+h^*)^{-1}})$ which will be the solution to the k-CAA problem.

It remains to compute the probability that $\mathcal{F}$ solves the k-CAA problem. Actually, $\mathcal{F}$ succeeds if:

Λ_1 : $\mathcal{F}$ does not abort during the simulation.
Λ_2 : σ^* is a valid forgery on (ID^*, PK_{ID^*}, m^*).
Λ_3 : $ID^* = ID_t$ and $c^* = 1$.

The advantage of $\mathcal{F}$ is

$$Adv_{\mathcal{F}}^{k-CAA} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2].$$

If Λ_1 happens, then

- $\mathcal{F}$ does not output $failure$ during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1 - \frac{1}{q_C})^{q_{VEx}}$.
- $\mathcal{F}$ does not output $failure$ during the simulation of signing oracle. This happens with probability $(1 - \frac{1}{q_C}\zeta)^{q_S} \geq (1-\zeta)^{q_S}$.

Consequently, $Pr[\Lambda_1] \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1-\zeta)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \frac{\zeta}{q_C}$. Therefore, $Adv_{\mathcal{F}}^{k-CAA} \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1-\zeta)^{q_S}\frac{\zeta}{q_C}\varepsilon$. The function $\zeta(1-\zeta)^{q_S}$ is maximized at $\zeta = \frac{1}{q_S+1}$. Therefore,

$$Adv_{\mathcal{F}}^{k-CAA} \geq (1 - \frac{1}{q_C})^{q_{VEx}}(1 - \frac{1}{q_S+1})^{q_S}\frac{1}{q_C(q_S+1)}\varepsilon.$$

This ends the proof □