パーソナルセキュリティモジュールの研究

課題番号：１３４５０１６２

平成13年度～平成15年度　科学研究費補助金

（基盤研究（B）（2））研究成果報告書

平成１６年3月

研究代表者　岡本 栄司
（筑波大学　電子情報・工学系）

はしがき

認証・暗号等に関する多くの機能を実現するパーソナルセキュリティモジュール（ＰＳＭ）を用い、各人がこれにより簡単かつ安全に取引や通信が行えるような仕組みを研究する。公開鍵暗号系を利用するが、その秘密鍵をＩＣカード等に格納しても、まだパーソナルセキュリティモジュールとしては不十分である。すなわち、暗号化メッセージを送る場合、それに先立って相手の公開鍵を入手する必要があるためである。これは、相手から貰うかあるいは公開鍵サーバ等から入手することになるが、入手した公開鍵が正しいものであるか否かの確証も必要になり面倒なものである。

そこで、ＩＤ情報に基づくセキュリティシステム（ＩD-Sec）がわが国を中心として長らく研究されてきた。これは、いわば名前そのものが公開鍵になっているようなものである。これならばわざわざ相手から公開鍵を貰うまでもない。しかしながら、公開鍵は勝手な数にすることができないため、まだ使いやすい方式は提案されていない。

そこで、本研究ではおもに方式の提案を行い、認証（署名）や鍵配送等の安全なプロトコルを検討した。ユーザＩＤは個別の秘密情報を持っているので、それを使って署名やユーザ識別子を作り、他人はそのＩＤを使ってチェックする形のマルチパーティプロトコルになる。楕円曲線上のPairingを用いた方式を検討し、安全性が高くて実用的な方式を提案した。

研究組織

研究代表者：岡本栄司（筑波大学　電子情報・工学系）
研究分担者：繆　いん（筑波大学　社会工学系）
研究分担者：岡本　健（筑波大学　電子情報・工学系）
研究分担者：満保雅浩（東北大学　情報シナジーセンター）
研究分担者：鄧　天波（東邦大学・理学部）
（研究協力者：George Davida）

交付決定額（配分額）　　　　　　　　　　（金額単位：千円）

	直接経費	間接経費	合　計
平成13年度	6,500	0	6,500
平成14年度	3,800	0	3,800
平成15年度	3,800	0	3,800
総　　計	14,100	0	14,000

研究成果による工業所有権の出願・取得状況

不正プログラム検出機能を持つシステム、及び方法
池田直樹、岡本栄司
特許出願番号　2004-030484
出願日　2004年2月17日

研究発表

(1) 学会誌等

1. Masakazu Soshi, Mamoru Maekawa and Eiji Okamoto; "The dynamic-typed access model and decidability of the safety problem", 電子情報通信学会英論文誌, Vol.E83, No. 1, pp.190-203, 2004
2. Tian-Bo Deng and Eiji Okamoto; "SVD-Based Design of Fractional-Delay 2-D Digital Filters Exploiting Specification Symmetries", IEEE Transactions on Circuits and Systems-I, Vol.50, No.8, pp.47-480, 2003
3. S. Russell, E. Okamoto, E. Dawson and J. Lopez; "Virtual certificates and synthetic certificates: New paradigms for improving public key validation", Computer Communication, Elsevier, Vol.26, No.16, pp.1826-1838, 2003
4. 金岡 晃, 岡本栄司; "IETF における侵入検知情報交換方式の問題点と対策", 情報処理学会論文誌、Vol.44, No.8, pp.1830-1837, 2003
5. 千石靖, 服部進実, 岡本栄司;"完全駆除までの期限を考慮したコンピュータウイルス駆除手法"、情報処理学会論文誌、Vol.44、No.1、pp.106-113, 2003
6. Tian-Bo Deng, Eriko Saito and Eiji Okamoto; "Efficient Design of SVD-Based 2D Digital Filters using Specification Symmetry and Order-Selecting Criterion", IEEE Trans. on Circuits and Systems-I, Vol.50, No. 2, pp.217-226, 2003
7. Sk. Md. M. Rahman, S. M. Masum, M. S. I. Khan, M. L. Rahman and E. Okamoto; Message digest and integrity checking using hash function, ICCIT2003 (6th International Conference on Computer and Information Technology), 2003
8. E. Dawson, J. Lopez, J. A. Montenegro and E. Okamoto; "BAAI: Biometric authentication and authorization infrastructure", Proc. of ITRE2003 (International Conference on Information Technology: Research and Education), IEEE, 2003
9. Takeshi Okamoto and Hirofumi Katsuno and Eiji Okamoto; "A Fast Signature Scheme Based on New On-line Computation", Proc. of ISC'03, LNCS, 2003
10. Juan M. Gonzalez Nieto, Jason Reid, Ed Dawson, Eiji Okamoto; "Privacy Implications of Trusted Computing", Trust and Privacy in Digital Business, DEXA workshop, IEEE, 2003
11. Akira Kanaoka and Eiji Okamoto; "Multivariate Statistical Analysis of Network Traffic for Intrusion Detection", Trust and Privacy in Digital Business, DEXA workshop, IEEE, 2003
12. E. Okamoto; "Cryptosystems Based on Polynomials over Finite Fields", Proc. of ITW (Information Theory Workshop), pp.74-77, 2002

13. E. Dawson, J. Lopez, J. A. Montenegro and E. Okamoto; "A new design of privilege management infrastructure for organizations using outsourced PKI", Proc. of ISC′02, LNCS 2433, Springer, pp.136-149, 2002
14. M. Mambo, E. Okamoto, Y. Sengoku and T. Tanaka; "Simple and secure access under the flexible organization management", Trust and Privacy in Digital Business, pp.456-460, Sep. 2-6, 2002
15. M. Henderson, R. Coulter, E. Dawson and E. Okamoto; "Modeling trust structures for public key infrastructures", Proceedings of ACISP′02, LNCS 2384, Springer, pp.56-70, 2002
16. E. Okamoto; "Permutation Polynomial and Its Application", Proc. of 2001 Workshop on Coding and Cryptography, Institute of Mathematical Science, National University of Singapore, 2001/9
17.

(2) 口頭発表

1. 岡本栄司; "我が国における暗号研究の発展について", 電子情報通信学会情報セキュリティ研究会予稿集, ISEC2003-109, pp.77-80, 2004
2. Mehdi Salah, Akira Kanaoka and Eiji Okamoto; "A Cooperative P2P Environment for Network Intrusion Detection", CSEC予稿集, 情報処理学会、2004
3. Routo Terada, Waldyr Dias Benits Jr. and Eiji Okamoto; "IBE Scheme to Exchange Authenticated Secret Keys", 電子情報通信学会 2004 暗号と情報セキュリティシンポジウム(SCIS2004)予稿集, 2004
4. 岡本 健, 髙木 剛, 岡本栄司; "Paringを用いた署名方式", SCIS2004予稿集, 2004
5. 樋口達巳, 岡本 健, 岡本栄司; "k-out-of-n 署名の拡張", SCIS2004 予稿集, 2004
6. 椎名信行, 岡本栄司, 岡本 健; "1-out-of-n 証明から k-out-of-n 証明への引き上げ方法", SCIS2004 予稿集, 2004
7. 金岡 晃, 岡本栄司; "侵入検知におけるパケットヘッダ情報の重要性", SCIS2004 予稿集, 2004
8. 西川弘幸, 太田良二, 八島一司, 千石 靖, 岡本栄司; "セキュリティホールを狙うワーム検出の実験", SCIS2004 予稿集, 2004
9. 菅原啓介, 千石 靖, 八島一司, 地下雅志, 西川弘幸, 岡本栄司; "未知ウイルス検出技術に関する一考察", SCIS2004 予稿集, 2004
10. 田中愛子, 小野 束, 岡本栄司; "ディジタル画像の改ざん検出用電子透かし法に関する一提案", SCIS2004 予稿集, 2004
11. 岩田 哲、大塚 玲、岡本栄司、岡本 健、川添 充、杉田 誠; "CRYPTO2003 会議報告"、電子情報通信学会信学技法 ISEC、2003

12. 伊藤忠彦、岡本栄司、岡本 健; "量子通信を利用した安全なカード配布に関する考察", CSS'03, 2003
13. 岡本 健、岡本栄司; "just" *k*-out-of-*n* 署名方式の提案、電子情報通信学会信学技法 ISEC2003-32、pp. 17-22, 2003
14. 田中愛子、金岡 晃、岡本栄司; "効率的な素数判定のための前処理について"、情報処理学会、CSEC 技術報告、20-25、pp. 139-142、2003
15. E. Okamoto, J. Lopez, E. Dawson, J. M. Gonzalez Nieto, S. Russell and J. Smith; Certificate Retrieval and Validation in Online Systems", Proc. of SCIS'03, 2B-1, pp.25-30, 2003
16. Juan Manuel Gonzalez Nieto, Jason Smith and Eiji Okamoto; "Securing Mobile IPv6 Binding Updates", Proc. of SCIS'03, 2B-4, pp.43-48, 2003
17. 金岡 晃、岡本栄司; "SePO (Self Protected Object)の提案とその実現方法について", CSS'02 予稿集, pp. 197-202, 2002
18. 金岡 晃、岡本栄司; "IDS 標準化: 実装と評価", CSS'02 予稿集, pp. 449-454, 2002
19. 金岡 晃、岡本栄司; "ニューラルネットワークを用いたデータマイニングによるリアルタイムネットワーク異常検出について", SCIS'02 予稿集, 9D-3, pp. 645-650, 2002 年 1 月
20. S. Russell, E. Okamoto and E. Dawson; "Improving performance in global PKI using virtual certificates and synthetic certificates", SCIS'02 予稿集, 15D-2, pp. 1149-1154, 2002 年 1 月
21. M. Burmester, Y. Desmedt, M. Mambo and E. Okamoto; "Formal modes and concrete examples of ordered multi-party cryptography", 電子情報通信学会情報信学技法セキュリティ研究会, 2001 年 12 月
22. T.-B. Deng and E. Okamoto; "Optimal design and parallel implementation of digital filters with variable magnitude and fractional-delay responses", 第 16 回ディジタル信号処理シンポジウム講演論文集, pp. 451-456, 2001 年 11 月
23. 土井 洋、岡本栄司、辻井重男; "擬似乱数列の衝突に関する一考察"、電子情報通信学会情報信学技法セキュリティ研究会 ISEC2001-7, pp. 45-48, 2001 年 5 月

(3) 出版物

1. 宮地、菊池監修、岡本他; 情報セキュリティ、IT Text シリーズ、オーム社、2003
2. 土居監修、内田、岡本、菊池、佐々木、寺田、村山編 ; 情報セキュリティ事典、共立出版、2003
3. 辻井重男、岡本栄司 ; 暗号のすべて、電波新聞社、2002
4. 岡本栄司; 暗号理論入門—改訂版—, 共立出版, 2002

社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

順序を考慮した暗号プロトコルのモデル化とその具体例

マイク・バーメスタ[†]　イボ・デスメット[†]　満保　雅浩[††]　岡本　栄司[†††]

[†] フロリダ州立大学計算機科学部
214 ラブ・ビルディング タラハセ 32306-4530
[††] 東北大学大学院情報科学研究科
980-8576 仙台市青葉区川内
[†††] 東邦大学理学部

E-mail: [†]{burmester,desmedt}@cs.fsu.edu, [††]mambo@icl.isc.tohoku.ac.jp, [†††]okamoto@sci.toho-u.ac.jp

あらまし　多くの暗号プロトコルにおいて、参加者の処理順序は重要な要素であるが、順序を考慮した方式ばかりが必ずしも検討されている訳ではない。そこで本論文では、暗号プロトコルにおける順序の重要性について着目し、考察を行う。まず、順序を考慮した暗号プロトコルの一般的なモデルを示した後に、幾つかの具体的な順序付暗号プロトコルを提案する。これらの提案プロトコルは、直観的に構成が可能な方式よりも効率的かつ能動的であり、しかも、暗号学的な条件付で、もしくは、暗号学的な条件なしで、順序に関する安全性を示すことができる。

キーワード　非対称性、順序付暗号プロトコル、閾値暗号プロトコル、順序付多重署名

Formal model of ordered multi-party cryptography and its concrete examples

Mike BURMESTER[†], Yvo DESMEDT[†], Masahiro MAMBO[††], and Eiji OKAMOTO[†††]

[†] Department of Computer Science, Florida State University
214 Love Building Tallahassee 32306-4530
[††] Graduate School of Information Sciences, Tohoku University
Kawauchi Aoba Sendai, 980-8576 Japan
[†††] Faculty of Science, Toho University

E-mail: [†]{burmester,desmedt}@cs.fsu.edu, [††]mambo@icl.isc.tohoku.ac.jp, [†††]okamoto@sci.toho-u.ac.jp

Abstract　With many multi-party applications the order of the participating parties is fundamental. However attention has mainly focused on schemes in which the order is irrelevant. In this paper we discuss the importance of order in multi-party cryptography. We give a formal model and present several ordered multi-party systems whose order is either unconditionally assured, or conditionally assured. The emphasis is on practical schemes which are efficient, dynamic and secure.

Key words　Asymmetry, ordered cryptosystems, threshold cryptosystems, ordered multi-signatures.

1. Introduction

In a multi-party application one can easily envisage circumstances in which the order of the participants is important. Typical examples are military and police operations where knowing the order of a command is crucial. It is important to know that the chain of command was followed. Take for example the case of a controversial order issued to a soldier by his commander. Even if the order by the commander may have been approved at a higher level, it is sometimes important to know whether the command was authorized before it was given, or whether the commander ignored the chain of command and went ahead giving a not yet approved order. Similar problems appear in the business world. The bankruptcy of Bairings Bank was a consequence of a subordinate bypassing the chain of command. In some circumstances the one who initiated an action is responsible, e.g. the one who ordered the murder of a competing mafia boss. In other cases the highest in rank is to blame when things go wrong, provided the chain of command has been followed. So, it may not be sufficient to know that all parties that should have been involved were involved, but also one must know *the order* in which they were involved.

Our work is related to tracing. We want to trace the order of the parties involved in a distributed computation. Today tracing is an important research area in cryptography. Examples of such papers are [2], [7]. The topic we study in this paper broadens the work already done. Our work allows to trace the order in which a multi-signature (or threshold signature) was signed. In case of dispute the evidence of the order may be used in court. It is clear that for a scheme to be acceptable, the length of the resulting multi-signature should be relatively short, the security should be high (at least proven secure), and that the scheme should be flexible, i.e., dynamic as we shall explain further down.

A lot of research has been done on multi-party protocols. We only list a few of a very long list of publications, e.g. [3], [5], [13], [14], [18]. For most of these multi-party protocols there is symmetry among the parties involved. For example with threshold signatures (in which a threshold number of parties can sign in the name of an organization) or with multi-signatures (in which several parties can make a joint signature using their own public key [18]), all the signing parties play the same role. The asymmetry is only between the receiver and those who signed the document. However there are many circumstances in which we need the group of signers to be ordered, i.e. the order of those involved in the protocol should be evident to an outsider.

To motivate the need for order in a multi-party protocol, we consider an example in the context of multi-signatures. Suppose that a proposal is to be submitted to a body, for example the board of directors of a company. Before submission, the proposal must be endorsed by a sufficient number of members of the organization. Often one wants to be able to trace the initiator of the proposal and the order in which the proposal is endorsed. One way to achieve this is by including in the proposal the list of endorsing parties. However this solution is not dynamic. It assumes that the list of parties is known (in an appropriate order) before the signing takes place. Many potential endorsers may not be willing to be included unless there are sufficiently many other parties who have already signed, in which case they may want their names to be in specified places (e.g. above or below others). An example of this is a petition: some parties will only endorse a petition if they are near the top of the list, and if sufficiently many others (perhaps of high ranking) have already endorsed it.

We note that there are different ways in which the order of a multi-party cryptosystem can be established. Consider for example a multi-signature of parties P_1, P_2, P_3. One way to assure this order is by designing a multi-signature which can only be verified sequentially by using the public keys of P_1, P_2, P_3, in this order. This does not necessarily mean that P_1, P_2, P_3 have signed in this order, only that this is the order established by the verification procedure. Even if the verification order reflects the signing order, the manipulation of the signing order may be feasible, and such a manipulation should be proven to be infeasible. Another way is to use a structured set of public keys which will assure the order. This is the approach that we shall use. We note that whichever way the order is established, what is important is that the parties are aware of this and that the order is assured.

It is sometimes argued that ordered cryptography cannot be convincingly implemented with public keys because the parties can "easily" modify the order by colluding. We claim that this argument is flawed. Clearly if all the parties involved agree, then they can generate another output with a different ordering. We cannot prevent this. However we shall show that there exist ordered cryptosystems for which it is computationally/unconditionally hard to modify the position of those parties who are not willing to co-operate. Furthermore the output of such systems uniquely determines the order of all parties (including any parties of a conspiracy).

As far as we know, the first to have observed the need for an ordering in cryptographic systems were Doi-Okamoto-Mambo-Uyematsu [10] in the context of multi-signatures. Some earlier schemes, as e.g. Okamoto's [18] multi-signature, also have properties that are order dependent. Recently, Mitomi-Miyaji proposed two multi-signatures that are "order flexible" and "order verifiable" (and "message flexible") [17]. Also Tada proposed an order-specified one-cycle multisignature scheme [22] which is secure against active attacks. The goal of our paper is to present a formal model of ordered multi-party cryptographic systems and to present practical examples whose order is proven.

Contribution: In the following section we consider an appropriate model for ordered multi-party protocols. Then, in Section 3. we present an unconditionally ordered multi-authentication scheme and a provably ordered multi-signature, which is rather inefficient. In Section 4. we consider efficient systems. We show how to make a provably ordered undeniable two-party signature scheme.

The main advantage of our efficient schemes (Sections 3.1 and 4.4) over the trivial scheme in Section 3.2 is that the size of the final multi-signature (multi-authenticator) is independent of the number of parties involved.

2. Model

In this section we introduce our model for ordered multi-party protocols.

2.1 Security model

We shall first describe our model informally. Let (P, V) be a two-party crypto-system and let E be the adversary (we do not exclude the possibility that $E = V$). We can think of P as the party that executes the protocol and V as the party that verifies the protocol. We shall assume that (P, V) satisfies the *completeness* conditions $\mathcal{C}$ involving P and V, and the *security* conditions $\mathcal{S}$ involving P, V and E. These conditions are determined by the requirements of the system.

Let $\mathcal{P}$ be a set of parties $P_1, P_2, \ldots, P_\ell$, and x an ordering of $\mathcal{P}$. Suppose that P in (P, V) is replaced by x, with the action of P replaced by the joint action of the parties in x (which use their secret key or their secret share). We denote the *transcript* of the execution of the distributed protocol (x, V) on input m by $T_{(x,V)}(m)$, and the output by $O_{(x,V)}(m)$. If no output is provided (to an outsider), then we take the transcript as output.

Let $\mathcal{P}' \subset \mathcal{P}$ be a set of dishonest parties who want to change the agreed order x. Obviously $\mathcal{P}'$ can change their own order. We say that the order of x in (x, V) is secure if it is hard for the dishonest parties to change the order of the honest parties, given the transcript $T_{(x,V)}(m)$ and the output $O_{(x,V)}(m)$. For example if $\mathcal{P} = \{P_1, P_2, P_3\}$, $\mathcal{P}' = \{P_1, P_3\}$ and $x = (P_1, P_2, P_3)$, then it should be hard for the parties in $\mathcal{P}'$ to change the order x in (x, V) to $x' = (P_2, P_1, P_3)$, given the transcript $T_{(x,V)}(m)$ and the output $O_{(x,V)}(m)$.

[Definition 1] (Sketch) - *Secure ordered multi-party cryptosystems*

The ordered multi-party cryptosystem (x, V) is *secure* if, on input (any) m, if the parties in x agree on this order, and if:

(1) (x, V) satisfies the completeness conditions $\mathcal{C}$ and the security conditions $\mathcal{S}$ of (P, V).

(2) It is computationally (unconditionally) hard for any subset $\mathcal{P}' \subset \mathcal{P}$ of dishonest parties to generate an output $O_{(x',V)}(m)$ for an ordering x'

of $\mathcal{P}$ in which the order of the honest parties $\mathcal{P}\backslash\mathcal{P}'$ is different, by using the transcript $T_{(x,V)}(m)$ (after observing the execution of (x,V)) and the output $O_{(x,V)}(m)$.
The dishonest parties in $\mathcal{P}'$ may deviate from protocol and are under the control of the adversary. The honest parties in $\mathcal{P}\backslash\mathcal{P}'$ adhere to the protocol (x,V).

There are several possible scenarios for the selection of the order x. For example this may be done by an *Initiator*. Alternatively it may be done in a distributed way by the parties in x. Thus P_{i_1} may select x, or just its successor P_{i_2}, or P_{i_2} and some of the other parties in x, and so on. For simplicity, we focus on the case when the input (message) m is selected by an Initiator I who will give it to the first party P_{i_1} of x (we may have $I = P_{i_1}$). However, this in general is not needed: the order can be established by using a multi-party protocol.

There are also several possible scenarios for the way in which the multi-party computation is executed by the authorized set x. This may involve a reliable *Combiner* C. This is the case when the protocol is executed in parallel. In this scenario, the Initiator I gives the message m to each party P_i of the set x. The parties P_i then execute their part of the multi-party protocol and send their output to the *Combiner*, which in turn computes the output of the protocol by combining appropriately the results it gets from the parties in x. The Combiner does not possess any secret keys, and its purpose is simply to compute its output from the (public) data output by the parties in x. Alternatively the multi-party computation may only involve the parties in x. This is the case when the protocol is executed serially. In this scenario, the first party P_{i_1} in x on input m from I executes its part of the multi-party protocol and then forwards its output to P_{i_2}. Then P_{i_2} executes its part of the protocol and forwards its output to P_{i_3}. And so on. Finally we can also have a combination of these scenarios (parallel and serial).

2.2 Reliability and robustness

In our definition we have not addressed the issue of reliability and robustness. The problem is that it may not be possible for the parties in $\mathcal{P}$ to agree on an ordering x for the multi-party protocol (x,V), given the input m. In the honest case (all parties in x adhere to the protocol) a dispute on a particular place in an ordering must be settled. However the dishonest parties may not agree to any such settlement (and insist on taking the place of an honest party). This would make it impossible to execute a valid ordered multi-party computation. Moreover, the dishonest parties may change their mind while the protocol is being executed.

If agreement on an order is possible then it is always possible to initiate a distributed multi-party computation by following the agreed order and bypassing those parties which do not adhere to the agreed order. It is not too difficult to see that the complexity of an ordered multi-party protocol, in particular an ordered multi-signature, with active adversaries, is the same as Byzantine agreement [11].

[Theorem 1] An ordered multi-party cryptographic system with active adversaries requires at least a protocol as complex as Byzantine agreement (and the same threshold of honest parties).

Proof. *Due to space limitation, proof is omitted.* ∎

We shall not discuss the agreement problem any further. In the rest of this paper we shall assume that for our multi-party protocols, on input m, agreement has been reached for a specific order x, and that the ordered system (x,V) is properly executed. Then, if (x,V) is secure, by Definition 1 it should be hard for dishonest parties to change the order of honest parties and produce an output $O_{(x',V)}(m)$.

3. Examples of ordered multi-party cryptosystems

3.1 An unconditionally ordered multi-authentication scheme

The notion of unconditionally secure (message) authentication (which is similar in many respects to unconditionally secure encryption using one-time keys) was originally introduced by Simmons in 1984 [21] Since then, several schemes and variants of these have been proposed. The main feature of these schemes is that, unlike digital signature schemes, the ability to authenticate messages is restricted to "insiders", that is to parties which possess some information not known to other parties.

As an illustration let us consider a well known unconditionally secure authentication scheme for which the message space is $\mathrm{GF}(q)$, the finite field with q elements. Let S be the sender and R the receiver (both "insiders"). The parties S, R share the one-time secret key pair (a, b), $a, b \in \mathrm{GF}(q)$. To authenticate a message $m \in \mathrm{GF}(q)$, the sender S sends R the pair (m, A), where $A = am + b$ is the *authenticator* of m. The receiver R checks the authenticity of m by using the secret key pair (a, b) to verify that $A \overset{?}{=} am + b$. It is easy to see that this scheme is unconditionally secure.

We now will show how to extend this basic scheme to get an *ordered* multi-authentication scheme. For simplicity we focus on linear access structures. Let $P_1, P_2, \ldots, P_\ell$ be the authenticating parties and let R be the receiver. Each party P_i shares with the receiver R a triple (a_i, b_i, c_i) with $a_i, b_i, c_i \in \mathrm{GF}(q)$, q sufficiently large, $q > \ell$. Suppose that the order in which the parties will authenticate a message $m \in \mathrm{GF}(q)$ is $P_{j_1}, P_{j_2}, \ldots, P_{j_\ell}$, where $j_1, j_2, \ldots, j_\ell$ is a permutation of $1, 2, \ldots, \ell$. Each party P_i computes the authenticator $A_i = a_i + b_i m + c_i k$ in $\mathrm{GF}(q)$, where $k \in \{1, 2, \ldots, \ell\}$ is the position of P_i in $P_{j_1}, P_{j_2}, \ldots P_{j_\ell}$. That is, k is such that $i = j_k$. The authenticator for message m is $A = \sum_{i=1}^{\ell} A_i$ computed in $\mathrm{GF}(q)$, which is sent to the receiver together with m and the authenticating order.

[Theorem 2] The probability that a dishonest party, say P'_{j_s} can claim successfully to be at location j_t, $j_t \neq j_s$, in the authentication sequence (without the help of P_{j_t}) is at most $1/q$.

Proof. *(Sketch) For simplicity, assume all parties except P_{j_t} conspire with P'_{j_s} in the attack. Then we get a set of equations with unknowns $a_{j_1}, \ldots, a_{j_t}, \ldots, b_{j_1}, \ldots, b_{j_t}, \ldots, c_{j_1}, \ldots, c_{j_t}, \ldots$, for which the values of $a_{j_1}, \ldots, a_{j_{t-1}}, a_{j_{t+1}}, \ldots, b_{j_1}, \ldots, b_{j_{t-1}}, b_{j_{t+1}}, \ldots, c_{j_1}, \ldots, c_{j_{t-1}}, c_{j_{t+1}}, \ldots$, and A_{j_t} are given (the other A_j's are linear combinations of the given values).*

Suppose now that the dishonest parties could create from these secret keys and the observed A_{j_t}, an authenticator $A'_{j_t} = a_{j_t} + b_{j_t} m + c_{j_t} t'$ with $j_t = j_{t'}$ and $t' \neq t$, for the desired permuted order. Then they could find c_{j_t} uniquely. However, the aforementioned equations give no information about c_{j_t}. So they must have guessed the correct c_{j_t} (with probability $1/q$). ∎

The proof can easily be extended to the case when dishonest parties claim more honest locations, and to the case when m is replaced at the same time.

3.2 A provably ordered multi-signatures – a trivial solution

Any cryptographic system can be trivially transformed into an ordered cryptosystem whose order is proven secure. Consider for example a signature scheme such as the DSS [9] (or El Gamal [12], RSA [20], etc.) Let the message space be M, let $P_1, P_2, \ldots, P_\ell$ be the authenticating parties, and let R be the receiver. Suppose that each party P_i has a public/private key pair (PK_i, SK_i), $i = 1, 2, \ldots, \ell$, and let the order in which the parties sign the message $m \in M$ be $P_{j_1} P_{j_2} \cdots P_{l_\ell}$. Each party P_i computes the signature $sign_{SK_i}(m, k)$, where $k \in \{1, 2, \ldots, \ell\}$ is the position of P_i in $P_{j_1} P_{j_2} \cdots P_{j_\ell}$. The multi-signature consists of all ℓ signatures.

It is straightforward to see that this scheme is as secure as the underlying primitive signature scheme, and that the same applies for its order. Furthermore, the scheme is not limited to signatures. Any primitive cryptosystem can be used to get a provably ordered multi-party variant. However this scheme is very inefficient. Indeed its communication and computational complexity grows proportionally to the number of parties involved, and is prohibitive for large groups of users.

In the following sections we will consider efficient solutions for provably ordered multi-party cryptosystems.

4. An efficient provably ordered undeniable multi-signature

4.1 Discussion

To start with suppose that there are only two parties P_1 and P_2 who are going to generate a joint undeniable multi-signature [6] and let $x = P_1 P_2$. Verifier V should be able to check that P_1 initiated the undeniable multi-signature when checking its validity. There are two natural cases.

Ordered undeniable multi-signatures, in which both parties have their own public key. In this case the receiver V should be able to verify who was the first to generate the undeniable signature scheme.

Ordered undeniable threshold signatures, in which the parties share a private key. In this case there is one public key and each party has a share of it. We require that only when the parties execute the multi-signature in the correct order will the output of the scheme be valid. We omit explaining this case in this paper.

We will generalize these schemes for more than two parties in Section 4.4. We first discuss our goals in greater detail.

4.2 An ordered undeniable two-party signature

For simplicity we explain our approach from the first viewpoint, and avoid discussing the details of the mathematics involved(1).

The set up

We use a discrete logarithm setting. As in [4], [6] we assume that p is an appropriate large prime, that $g \in Z_p$ has order a large prime q and that p, q and g are public. The private key of P_1 is the ordered pair (x_{11}, x_{12}), $x_{11}, x_{12} \in_R Z_q^*$ and the public key is (y_{11}, y_{12}), where $y_{11} = g^{x_{11}}$, $y_{12} = g^{x_{12}}$, both evaluated in $\langle g \rangle$ (the subgroup generated by g). Similarly P_2's private key is (x_{21}, x_{22}), $x_{21}, x_{22} \in_R Z_q^*$, and the public key is (y_{21}, y_{22}) with $y_{21} = g^{x_{21}}$, $y_{22} = g^{x_{22}}$.

All operations are evaluated in the subgroup $\langle g \rangle$ of Z_p^*: when there is no ambiguity we will not state this explicitly.

Generating a two-party signature

Consider the case when the authorized sequence is P_1P_2. Note that it is easy to check that $\text{ord}(m) = q$ by testing that $m \neq 1 \bmod p$ and $m^q = 1 \bmod p$. Let $m \in \langle g \rangle$ be the input (message).

Step 1 Each party P_i, $i = 1, 2$, checks that $\text{ord}(m) = q$. If so, P_i computes the partial signature $s_i(m) = m^{x_{ii}}$. Then P_i sends $s_i(m)$ to a Combiner C.

Step 2 C checks that $\text{ord}(s_i(m)) = q$ for $i = 1, 2$. If so, C computes $s_{12}(m) = s_1(m) \cdot s_2(m) = m^{x_{11}+x_{22}}$. This is the multisignature of m by x. C sends $s_{12}(m)$ to the Verifier V.

This is a parallel implementation. For a sequential implementation the Combiner is not needed. Party P_1 sends the partial signature $s_1(m) = m^{x_{11}}$ to P_2 who, after checking the correctness of $s_1(m)$, computes $s_{12}(m)$, and sends this to V. Observe that if P_2 would have been first and P_1 second, then the resulting undeniable signature would be $s_{21}(m) = m^{x_{21}+x_{12}}$.

Confirmation of a two-party signature

The goal of the confirmation protocol is to prove that a two-party signature is valid. This is based on Chaum's confirmation protocol [4].

To confirm that $s_{12}(m)$ is the correct two-party signature of m by P_1P_2, the parties V, C, P_1, P_2 proceed as follows.

Step 1 The Verifier V checks that $\text{ord}(s_{12}(m)) = \text{ord}(m) = q$. If so, V selects $a, b \in_R Z_q$ and computes $u = m^a g^b$. V sends u to the Combiner C who forwards it to P_1, P_2.

Step 2 Each P_i, $i = 1, 2$, chooses $r_i \in_R Z_q$, computes $w_i = u^{x_{ii}} \cdot g^{r_i}$, and sends it to C.

Step 3 C computes $w = w_1 \cdot w_2$ and sends this to V.

Step 4 V sends a, b to C who forwards these to P_1, P_2.

Step 5 Each P_i, $i = 1, 2$, checks that u was correctly constructed. If so, P_i sends r_i to C.

Step 6 C computes $r = r_1 + r_2 \bmod q$ and sends this to V.

Step 7 V computes $v = g^r$ and then w/v, and checks that this is the same as $s_{12}(m)^a \cdot (y_{11}y_{22})^b$. If so, V accepts $s_{12}(m)$ as correct.

This is a parallel implementation. For a sequential implementation, P_2 acts as a Combiner.

(1): Our scheme corresponds to an almost module with two external operators, which do not form a bi-module [15].

Disavowal of an invalid two-party signature

The goal of the disavowal protocol is to show that a forged two-party signature z is invalid. Our protocol is based on Chaum's disavowal protocol [4].

Let k be a sufficiently small number, let status$(z) \in \{0, valid, forged, unknown\}$ and let $t \in Z_k$. To confirm that z is not a valid multi-signature of m by P_1P_2, the parties V, C, P_1, P_2 proceed as follows.

Step 1 The Verifier V checks that $\mathrm{ord}(z) = \mathrm{ord}(m) = \mathrm{ord}(y_{11}) = \mathrm{ord}(y_{22}) = q$. If so, V selects $a \in_R Z_q$ and $s \in_R Z_k$ and computes $u = m^s g^a$ and $v = z^s y^a$, where $y = y_{11}y_{22}$. V sends u, v to P_1, P_2.

Step 2 For $t = 1, 2, \ldots, k-1$:
P_1 chooses $r'_{1t}, r''_{1t} \in_R Z_p$ and computes $A_t = r'_{1t} u^{x_{11}} \bmod p$, $B_t = r''_{1t} m^{x_{11}t} \bmod p$ and sends these to P_2. P_2 chooses $r_{2t} \in_R Z_p$ and computes:

$$\begin{aligned} w_t &= r_{2t}[(A_t u^{x_{22}} z^t)/(B_t m^{x_{22}t} v)] \bmod p \\ &= r_{1t} r_{2t} \, (m^x/z)^{s-t} \bmod p, \end{aligned}$$

where $r_{1t} = r'_{1t}/r''_{1t} \bmod p$ and $x = x_{11} + x_{22} \bmod q$.

Step 3 P_1 computes $D_t = g^{r_{1t}} \bmod p$, $t = 1, 2, \ldots, k-1$, commits to $\{D_t\}$, and sends the commitments to C and V.
P_2 computes $w_{1t} = w_r/r_{2t} \bmod p$ and $E_t = g^{w_{1t}} \bmod p$, $t = 1, 2, \ldots, k-1$, and then commits to $\{E_t\}$, and sends the commitments to C and V.

Step 4 V sends a, s to P_1 and P_2.

Step 5 P_1, P_2 check that u and v are properly constructed. If so, they open their commitments to C and V.

Step 6 C checks that $\mathrm{ord}(D_t) = \mathrm{ord}(E_t) = q$, $t = 1, 2, \ldots, k-1$. Set status$(z) := 0$, $t := 0$.
Case $D_t = E_t$: If status$(z) = 0$, set status$(z) := unknown$ and $t := t+1$; if status$(z) = unknown$, set status$(z) := valid$, and output $s \in_R Z_k$; if status$(z) = forged$, output $s = t$.
Case $D_t \neq E_t$: If status$(z) = unknown$, set status$(z) := forged$, and output $s = 0$; else set status$(z) := forged$, $t := t + 1$.

Step 7 V accepts that z is not valid if the opened value of s is correct.

P_1, P_2 can cheat with probability $1/k$. To reduce this, the protocol is repeated a few times (independently).

The subtle issues addressed by Jacobson-Yung [16] will be discussed in the final paper.

4.3 Security: the two-party case

We now prove that our undeniable two-party signature scheme satisfies our goals for ordered systems, assuming that the Diffie-Hellman decision problem over the subgroup $\langle g \rangle$ of Z_p^* is hard. That is, given (p, g, g^x, g^y, h), it is hard to decide if $h = DH_g(g^x, g^y)$, where $DH_g(g^x, g^y) = g^{xy} \bmod p$. The Diffie-Hellman (search) problem is to compute $DH_g(g^x, g^y)$ given (p, g, g^x, g^y) [8]. Clearly if the Diffie-Hellman decision problem is hard then so is the Diffie-Hellman (search) problem.

[Theorem 3] If the Diffie-Hellman decision problem in the subgroup $\langle g \rangle$ of Z_p^* is hard, then the undeniable two-party signature scheme achieves our order goals.

Proof. *Due to space limitation, proof is omitted.* ∎

4.4 Ordered undeniable multi-signatures: the general case

We now extend our scheme to the general case. As in the two-party case we use a discrete logarithm setting and work in the subgroup $\langle g \rangle$ of Z_p^* of prime order q. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_\ell\}$, $\ell = O(\log |p|^c)$, c constant, be the set of parties. Each party P_i has $\lceil \log_2 \ell \rceil$ private keys: $x_{ij} \in_R Z_q^*$, $j = 1, 2, \ldots, \lceil \log_2 \ell \rceil$, and as many public keys: $g^{x_{ij}}$.

Let $x = P_{j_1} P_{j_2} \cdots P_{j_t}$ be an ordering of the parties. To determine the partial signature of P_{j_i} we use the private keys of P_{j_i} which correspond to its position in x. That is, the private keys which correspond to the non-zero entries in the binary representation $b_{i1} b_{i2} \cdots b_{i\lceil \log_2 t \rceil}$ of i (we take $i = \sum_{k=1}^{\lceil \log_2 t \rceil} 2^{k-1} b_{ik}$). The partial signature of m by P_{j_i} is thus $s_{j_i}(m) = m^{\sum_{k=1}^{\lceil \log_2 t \rceil} b_{ik} x_{j_i k}}$, and the multi-signature of m by

$P_{j_1}P_{j_2}\cdots P_{j_t}$ is:

$$s_{i_1 i_2 \cdots i_{\lceil \log_2 t \rceil}}(m) = m^{\left(\sum_{i=1}^{t}\sum_{k=1}^{\lceil \log_2 t \rceil} b_{ik}x_{j_i k}\right)}.$$

It is easy to see how to extend the multisignature generation protocol. Each P_{j_i} sends $s_{j_i}(m)$ to the Combiner C, and C multiplies these to get $s_{i_1 i_2 \cdots i_{\lceil \log_2 t \rceil}}(m)$. Both the confirmation and disavowal protocols can be extended to the general case (we postpone this for the final paper).

[Theorem 4] If the Diffie-Hellman decision problem in the subgroup $\langle g \rangle$ of Z_p^* is hard, then the undeniable multi-signature scheme in Section 4.2 achieves our goals for ordered systems (Definition 1).

Proof. *This is an extension of the two-party case. Details will be given in the final paper.*

∎

In order to make the order of honest parties secure, the method for the general case described above is applicable to ElGamal-type multisignature schemes in general.

5. Conclusion

In this paper we have discussed the importance of order (asymmetry) in multi-party cryptography. We presented a formal security model for order and proposed unconditionally ordered authentication systems and provably ordered multi-signatures. These systems are simple and efficient.

References

[1] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. Nat. Computer Conf. AFIPS Conf. Proc.*, pages 313–317, 1979. vol.48.

[2] D. Boneh and M. Franklin. An efficient public key traitor tracing scheme. In M. Wiener, editor, *Advances in Cryptology — Crypto '99, Proceedings (Lecture Notes in Computer Science 1666)*, pp. 338–353. Springer-Verlag, 1999.

[3] C. Boyd. Digital multisignatures. In H. Beker and F. Piper, editors, *Cryptography and coding*, pp. 241–246. Clarendon Press, 1989. Royal Agricultural College, Cirencester, December 15–17, 1986.

[4] D. Chaum. Zero-knowledge undeniable signatures. In I. Damgård, editor, *Advances in Cryptology, Proc. of Eurocrypt '90 (Lecture Notes in Computer Science 473)*, Springer-Verlag, pp. 458–464, 1991.

[5] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 11–19, May 2–4, 1988.

[6] D. Chaum and H. van Antwerpen. Undeniable signatures. In G. Brassard, editor, *Advances in Cryptology – Crypto '89, Proceedings (Lecture Notes in Computer Science 435)*, Springer-Verlag, pp. 212–216, 1990.

[7] B. Chor, A. Fiat, and M. Naor. Tracing traitors. In Y. G. Desmedt, editor, *Advances in Cryptology — Crypto '94, Proceedings (Lecture Notes in Computer Science 839)*, pp. 257–270. Springer-Verlag, 1994.

[8] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT–22(6) pp. 644–654, 1976.

[9] A proposed federal information processing standard for Digital Signature Standard (DSS). Federal Register, 1991.

[10] H. Doi, E. Okamoto, M. Mambo and T. Uyematsu. Multisignature Scheme with Specified Order. SCIS94, Symposium on Cryptography and Information Security, Biwako, Japan, January 27-29, 1994.

[11] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, January 1993.

[12] T. ElGamal. A Public Key Cryptosystem and a Signature scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory*, IT-31(4), pp. 469–472, 1985.

[13] Y. Frankel. A practical protocol for large group oriented networks. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology, Proc. of Eurocrypt '89 (Lecture Notes in Computer Science 434)*, Springer-Verlag, pp. 56–61, 1990.

[14] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth annual ACM Symp. Theory of Computing, STOC*, pp. 218–229, 1987.

[15] N. Jacobson. *Basic Algebra II.* W. H. Freeman and Company, New York, 1989.

[16] M. Jakobsson and M. Yung. Proving without knowing. In N. Koblitz, editor, *Advances in Cryptology — Crypto '96, Proceedings (Lecture Notes in Computer Science 1109)*, Springer-Verlag, pp. 186–200, 1996.

[17] S. Mitomi and A. Miyaji. A general model of multisignature schemes with message flexibility, order flexibility and order verifiability. IECE TRans. Fundamentals, E84-A (10), pp. 2488–2499, 2001,

[18] T. Okamoto. A Digital Multisignature Scheme Using Bijective Public-Key Cryptosystems. *ACM Trans. on Comp. Systems*, 6(8), pp. 432–441, 1988.

[19] K. Ohta and T. Okamoto. A digital multisignature scheme based on the Fiat-Shamir scheme. In H. Imai, R.L. Rivest, T. Matsumoto, editors, *Advances in Cryptology, Proc. of Asiacrypt '91 (Lecture Notes in Computer Science 739)*, pp. 139–148. Springer-Verlag, 1993.

[20] R.L. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of ACM*, 21(2), pp. 120–126, 1978.

[21] G.J. Simmons, Authentication theory/Coding theory. In G.R. Blakeley and D. Chaum, editors, *Advances in Cryptology, Proc. of Crypto '84 (Lecture Notes in Computer Science 196)*, Springer-Verlag, pp. 411–431, 1985.

[22] M. Tada, An order-specified one-cycle multisignature scheme secure against active attacks, CSS, Computer Security Symposium, IPSJ Symposium series, 2001(15), pp. 217–222, 2001.

Cryptosystems Based on Polynomials over Finite Fields

Eiji Okamoto
Information Sciences and Electronics,
University of Tsukuba,
305-8573 Japan
e-mail: okamoto@is.tsukuba.ac.jp

Abstract — Two cryptosystems using polynomials on finite field are introduced; permutation cipher based on permutation polynomials and flexible secret sharing scheme. Polynomials are called permutation polynomials if they induce bijective functions. Metric and algebraic properties of permutation polynomials over a finite field are investigated, especially properties associated with orbits and permutation cycles. Flexible secret sharing schemes have features that they can change a secret, enroll/disenroll members, increase/decrease threshold, without changing shares. Simple scheme using polynomials over finite field is presented.

I. Introduction

Some cryptosystems using polynomials over finite field have been proposed so far. Dickson scheme, C* public key cryptosystem are examples of such cryptosystems. RSA cryptosystem is an encryption transformation based on polynomials of high degree. Another example of cryptosystems using polynomials is elliptic curve cryptosystems, because they are constructed on polynomials over finite field.

In this paper we introduce two cryptosystems using polynomials, which are not well known yet. A permutation cipher and a secret sharing scheme.

II. Permutation ciphers using permutation polynomials

Permutations are realized by random access memory(RAM) and two address generators. Suppose the RAM has n addresses and is linked to Address Generators 1 and 2. Address Generator 1 and 2 generate addresses for writing and reading, respectively. Samples are written according to the addresses generated by Address Generator 1. Samples are read out according to the addresses generated by Address Generator 2. After writing n samples, reading of n samples is carried out. In the permutation descrambler, Address Generators 1 and 2 are interchanged. No inverse transformation of the permutation is required.

One of the two address generators, say Address Generator 2, generates a random address sequence which is varied according to an encryption key. Address Generator 1 generates a fixed address sequence. The permutation scheme's security depends on the address sequences. For example, if the addresses generated by Address Generators 1 and 2 are similar, the scrambled signal is almost the same as the original signal.

In this paper, Address Generators 1 and 2 are constructed using polynomials $f_K(x)$ over $GF(q)$, $q = p^m$, or their iterations. Here m is the length of an address "digit" list and p is a prime number. Since every element of $GF(q)$ can be represented by a list of m "digits", addresses can be regarded as elements of $GF(q)$. Assume, without loss of generality, that Address Generator 1 generates

$$0, 1, \alpha, \alpha^2, \cdots, \alpha^{q-2}, 0, 1, \alpha, \alpha^2, \cdots$$

where α indicates a primitive element in $GF(q)$[1]. And assume that Address Generator 2 generates

$$f_K(0), f_K(1), f_K(\alpha), f_K(\alpha^2), \cdots, f_K(\alpha^{q-2}), f_K(0), \cdots,$$

using a polynomial $f_K(x) = a_0 + a_1x + \cdots + a_dx^d$. Here K denotes an encryption key which consists of the coefficients list

$$K = (a_0, a_1, \cdots, a_d). \tag{1}$$

In other words, changing keys in this cipher means changing coefficients of the polynomial.

$f_K(x)$ should be a bijective function from $GF(q)$ to $GF(q)$. Moreover, $f_K(x)$ has to be nonlinear, that is $d > 1$, from a security point of view. We exhibit bijective polynomials, called permutation polynomials [3], over $GF(q)$. Any bijective functions could be used instead of the $f_K(x)$, but polynomials have several good features, described below from the standpoint of randomness and security criteria.

Theorem 1 *The permutation ciphers based on the $f_K(x)$ satisfy the conditions below.*

1. *If $d > 1$, no more than d samples are fixed points of the permutation. In other words, there are no more than d elements x satisfying*

$$f_K(x) = x \tag{2}$$

2. *In the decoded sequence, with a decryption key K_2 which is different from the encryption key K_1, fewer*

[1] We could change Address Generator 1 in such a fashion that it generates $(1, \alpha, \alpha^2, \cdots, \alpha^{q-2}, 1, \alpha, \alpha^2, \cdots)$. However, constant a_0 should be 0 in this case. This scheme offers simpler ciphers, and Theorem 1 could be stated much simpler, but the key size becomes smaller.

than d samples are fixed points. In other words, there are no more than d elements x satisfying

$$f_{K_2}^{-1}(f_{K_1}(x)) = x, \tag{3}$$

where f_K^{-1} denotes the inverse of the permutation f_K, and $K_1 \neq K_2$.

3. *Let $a_d = 1$. In a permuted and shifted sample sequence, no more than $2d+2$ samples are fixed points. In other words, for any fixed k ($1 \leq k \leq q-2$), there are no more than $2d+2$ elements x satisfying*

$$f_K(x) = \begin{cases} \alpha^{k-1} & x = 0 \\ \alpha^k x & x = \alpha^i, i+k \leq q-2 \\ 0 & x = \alpha^i, i+k = q-1 \\ \alpha^{k-1}x & x = \alpha^i, i+k \geq q. \end{cases} \tag{4}$$

4. *Suppose that $d > 1$, $\gcd(d, q-1) = 1$, and $a_d = 1$. Then in a sequence which is inversely permuted with any key and shifted, no more than $2d+2$ samples are fixed points. In other words, for any fixed k ($1 \leq k \leq q-2$), there are no more than $2d+2$ elements x satisfying*

$$f_{K_2}^{-1}(f_{K_1}(x)) = \begin{cases} \alpha^{k-1} & x = 0 \\ \alpha^k x & x = \alpha^i, i+k \leq q-2 \\ 0 & x = \alpha^i, i+k = q-1 \\ \alpha^{k-1}x & x = \alpha^i, i+k \geq q. \end{cases} \tag{5}$$

Polynomial Representing Permutation over $GF(2^m)$ and Orbits

To characterize permutation polynomials, we introduce three orbits on the set of polynomials. Let q be p^m. We concentrate mainly on the case $p = 2$. This is generally considered the most difficult case, and in almost all applications, $p = 2$.

Consider linear polynomials: $\phi \mapsto a\phi + b$ where $(a, b) \in (GF(q) - \{0\}) \times GF(q)$. These form a subgroup L. L acts on a polynomial f on left, right or both and produce a left orbit Lf, a right orbit fL or a double orbit LfL. These orbits induce equivalence relations, respectively.

Definition 1 *Suppose the set of linear polynomials be L. A left orbit Lf, a right orbit fL and a double orbit LfL of a polynomial f are*

$$Lf = \{af(x) + b | (a, b) \in GF(q)^- \times GF(q)\} \tag{6}$$

$$fL = \{f(\alpha x + \beta) | (\alpha, \beta) \in GF(q)^- \times GF(q)\} \tag{7}$$

$$LfL = \{af(\alpha x + \beta) + b | (a, b), (\alpha, \beta) \in GF(q)^- \times GF(q)\}. \tag{8}$$

Here $GF(q)^-$ shows $GF(q) - \{0\}$.

The size of each left orbit and each right orbit is $q(q-1)$.

Orbits are almost 'orthogonal to the metric structure' of corresponding permutations, as the next two theorems indicate. Before stating the theorem, a metric $dist(f, g)$ is introduced in $GF(q)[x]$.

Definition 2 *Hamming distance $dist(f, g)$ between $f, g : GF(q) \mapsto GF(q)$ is defined by setting*

$$dist(f, g) = |\{x \in GF(q) : f(x) \neq g(x)\}|. \tag{9}$$

Let $P_\delta = \{f(x) \in GF(q)[x] | deg(f(x)) \leq \delta,$ $f(x)$ is a permutation polynomial$\}$ for $\delta < q$.

Theorem 2
$$dist(f, g) \geq q - \delta \tag{10}$$
if $f, g \in P_\delta$ and $g \neq f$. Hence if $g \in LfL$ and $g \neq f$, then
$$dist(f, g) \geq q - \delta. \tag{11}$$

Theorem 3 *$dist(f, g) \geq q - 1$ if $g \in Lf$ or $g \in fL$, and $g \neq f$. Equality holds if and only if $g(x) = f(\alpha x + \beta), \alpha \neq 1$ or $g(x) = af(x) + b, a \neq 1$.*

Polynomials Corresponding to Permutation Cycles

We discuss algebraic cycle structures of permutations, especially, *fixed points*, *derangements*, *transpositions*, *cycles* and *products of cycles*, because a permutation is composed of pairwise disjoint cycles. A permutation polynomial corresponding to π is written as $f_\pi(x)$ in this paper.

Fixed Point and Derangement

Fixed points are cycles of length 1 and derangements are permutations which have no fixed points.

Theorem 4 *The degree of a non-identity permutation polynomial of degree 1 or more with k fixed points is at least k.*

The probability that a permutation is a derangement is $1/e \approx 0.37$. Not all permutation polynomials are derangements, but Theorem 4 shows that the probability that a given x satisfies $f(x) = x$ is small if the degree δ of $f(x)$ is small.

Transposition

Any permutation can be expressed as a product of transpositions. A transposition is a permutation which interchanges two elements.

Theorem 5 *Assume a transposition permutes r_1, r_2, and let $r_3, r_4, \cdots, r_q$ be other (fixed) elements of $GF(q)$. Then, the polynomial corresponding to a transposition (r_1, r_2) is represented by*

$$g(x) = (r_1 - r_2)^2(x - r_3)(x - r_4)\cdots(x - r_q) + x. \tag{12}$$

Cycle and Product of Cycles

If $\pi(r_i) = r_{i+1}$ $(i = 1, 2, \cdots, k-1)$, $\pi(r_k) = r_1$, $r_i \neq r_j$ $(i \neq j)$, then π is called (algebraic) cycle and written as $(r_1 r_2 \cdots r_k)$. Any permutation is represented by a product of disjoint cycles. A polynomial corresponding to a cycle can be given explicitly.

Let $h_{a,b}(x)$ be a polynomial of degree $q-2$ defined by

$$h_{a,b}(x) = \prod_{r \neq a,b} (x - r). \tag{13}$$

Then, $h_{a,b}(a) = h_{a,b}(b) = (a-b)^{-1}$, $h_{a,b}(r) = 0$, for any $r \neq a, b$.

Theorem 6 *A polynomial corresponding to the cycle $C = (r_1\ r_2\ \cdots\ r_n)$ is*

$$f_C(x) = \sum_{i=1}^{n-1} (r_1 - r_{i+1})(r_i - r_{i+1}) h_{r_i, r_{i+1}}(x) + x. \tag{14}$$

Note that the degree of $f_C(x)$ is not necessarily of degree $q-2$.

A general form of a permutation polynomial $f_C(x)$ corresponding to a product $C = C_1 C_2 \cdots C_k$ of pairwise disjoint cycles is given by the next theorem.

Theorem 7 *A polynomial representation of a product $C = C_1 C_2 \cdots C_k$ is*

$$f_C(x) = \sum_{i=1}^{k} (f_{C_i}(x) - x) + x. \tag{15}$$

Distribution of the Degrees of Permutation Polynomials

The general form of permutation polynomials corresponding to a cycle or a product of pairwise disjoint cycles are shown in Theorems 6, 7. However, they do not indicate the actual degree except the case of transpositions. Theorem 4 shows that the degree of a permutation polynomial corresponding to a cycle or a product of pairwise disjoint cycles is at least $q-k$, where k is the total length of the cycle or the product of pairwise disjoint cycles when $k > 1$.

However, it is difficult to investigate the exact degree of permutation polynomials corresponding to cycles in general. In Table 1, the exact degrees of permutation polynomials corresponding to a small cycle or a product of pairwise disjoint small cycles are shown.

III. Flexible secret sharing scheme

Secret sharing scheme was proposed by G. R. Blakley and A. Shamir independently. Blakley used hyperplanes and Shamir used polynomials on a finite field. In this paper Shamir's (k, n) scheme is adopted but Blakley scheme could be used also. Here n is the number of users and k is threshold.

In a real situation, it happens that secret information should be modified. However, in this case, all shares must be changed and this is not considered to be realistic. To overcome this difficulty, we introduce a flexible secret sharing scheme using polynomials on a finite field.

Shamir's scheme

Shamir's scheme picks up $k-1$ coefficients randomly and makes a polynomial

$$f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} \tag{16}$$

with a constant of the secret s, and gives $S_i = f(u_i)$ to user i as a share. u_i is an identifier of user i.

Any k users can recover the secret s, because k equations

$$s + a_1 u_i + a_2 u_i^2 + \cdots + a_{k-1} u_i^{k-1} \tag{17}$$
$$i = i_1, i_2, \cdots, i_{k-1}$$

can determine k unknowns, in particular the secret s.

Flexible scheme

Consider the case that the secret s is changed to new one s'. In this flexible scheme, all shares given first, then a polynomial $g(x)$ is determined as to cross all shares and the new secret.

The degree of polynomial $g(x)$, n, is usually much higher than the degree of original polynomial $f(x)$, k.

$$g(x) = s' + b_1 x + b_2 x^2 + \cdots + b_n x^n \tag{18}$$

Hence, any k users cannot recover secret s'.

To overcome this difficulty, $n-k+1$ coefficients are made public. Then the number of unknowns is $(n+1) - (n-k+1) = k$, and any k users can recover the secret s' from the equations below. Higher coefficients are public.

$$s' + b_1 u_i + b_2 u_i^2 + \cdots + b_{k-1} u_i^{k-1}$$
$$= S_i - b_k u_i^k - \cdots - b_n u_i^n \tag{19}$$
$$i = i_1, i_2, \cdots, i_{k-1}$$

We need a trusted "dealer" who decides $f(x)$ to distribute shares, and calculates the new polynomial $g(x)$. However, the dealer does not have to get all shares and the new secret, nor calculate $g(x)$. His job at updating is to calculate higher $n-k+1$ coefficients to publish. To do this, he has only to know

$$\delta, u_1, u_2, \cdots, u_n,$$

here $\delta = s' - s$ and u_i is the identifier of user i. Then he can calculate polynomial $h(x)$ which crosses

$$(\delta, 0), (u_1, 0), (u_2, 0), \cdots, (u_n, 0).$$

This polynomial $h(x)$ must be a polynomial of degree n:

$$h(x) = \delta + c_1 x + c_2 x^2 + \cdots + c_n x^n \tag{20}$$

Two polynomials $g(x)$ and $h(x) + f(x)$ are identical, because both of the polynomials cross $n+1$ points $(0, s'), (u_1, f(u_1)), (u_2, f(u_2)), \cdots, (u_n, f(u_n))$ and both degrees are same, n.

Higher $n-k+1$ coefficients of $h(x)$, $c_k, c_{k+1}, \cdots, c_n$, made to be public. These coefficients are same as

Table 1: Distribution of permutation polynomials of various cycle structures, $q = 2^m > 4$

cycle(s)	deg $q-2$	deg $q-3$	deg $q-4$	deg $q-5$	note
2-cycle	1	0	0	0	
3-cycle	1	0	0	0	m odd
	$\frac{q-4}{q-2}$	$\frac{2}{q-2}$	0	0	m even
4-cycle	$\frac{q^2-7q+14}{(q-2)(q-3)}$	$\frac{2(q-6)}{(q-2)(q-3)}$	$\frac{4}{(q-2)(q-3)}$	0	$m \equiv 0 \bmod 4$
	1	0	0	0	$m \equiv 1, 3 \bmod 4$
	$\frac{q^2-7q+14}{(q-2)(q-3)}$	$\frac{2(q-4)}{(q-2)(q-3)}$	0	0	$m \equiv 2 \bmod 4$
2-2 permutation	$\frac{q-4}{q-3}$	0	$\frac{1}{q-3}$	0	
2-3 permutation	$\frac{q^2-6q+12}{(q-2)(q-3)}$	$\frac{(q-6)(q-7)}{(q-3)(q-4)}$	$\frac{3(q-8)}{(q-2)(q-3)(q-4)}$	$\frac{6}{(q-2)(q-3)(q-4)}$	$m \equiv 0 \bmod 6$
	$\frac{q^2-8q+18}{(q-3)(q-4)}$	$\frac{q-6}{(q-3)(q-4)}$	0	0	$m \equiv 1, 5 \bmod 6$
	$\frac{q^2-6q+12}{(q-2)(q-3)}$	$\frac{q-6}{(q-2)(q-3)}$	0	0	$m \equiv 2, 4 \bmod 6$
	$\frac{q^2-8q+18}{(q-3)(q-4)}$	$\frac{(q-5)(q-6)}{(q-3)(q-4)}$	$\frac{3(q-8)}{(q-2)(q-3)(q-4)}$	$\frac{6}{(q-2)(q-3)(q-4)}$	$m \equiv 3 \bmod 6$

$b_k, b_{k+1}, \cdots, b_n$ respectively, because coefficients of degree k or higher of $f(x)$ are all zero.

Any k users can recover the new secrets s' in the same way in the previous discussion:

$$\begin{aligned} s' + b_1 u_i + b_2 u_i^2 + \cdots + b_{k-1} u_i^{k-1} & \\ = S_i - c_k u_i^k - \cdots - c_n u_i^n & \qquad (21) \\ i = i_1, i_2, \cdots, i_{k-1} & \end{aligned}$$

This means k users can recover $g(x)$ directly, not through $h(x)$.

References

[1] L. E. Dickson: "The Analytic Representation of Substitutions on a Power of a Prime Number of Letters with a Discussion of the Linear Group", Annals of Mathematics, vol.11, pp.65-120, pp.161-183, 1896-1897

[2] H. Imai: "Information Security Aspects of Spread Spectrum Systems", Advances in Cryptology - Proc. of Asiacrypt'94, Springer, pp.195-208, 1994

[3] R. Lidl and H. Niederreiter: "Finite Fields", Addison-Wesley, 1983

[4] G. L. Mullen: "Permutation Polynomials over Finite Fields", finite field, coding theory, and advances in communications and computing, lecture notes in pure and applied mathematics, 144, edited by G. L. Mullen and P. J. Shiue, marcel dekker, 1993

[5] E. Okamoto, W. Aitken, G. R. Blakley and I. Borosh: "Properties of Permutation Polynomials", Proc. 3rd Int'l Conf. Finite Fields and Applications, 1995

[6] E. Okamoto, W. Aitken and G. R. Blakley: "Algebraic Properties of Permutation Polynomials", IEICE Trans. on Fundamentals, vol.E79-A, no.4, pp.494-501, 1996

[7] E. Okamoto, W. Aitken, G. R. Blakley and P. F. Stiller: "Simple Permutation Ciphers Using Polynomials over a Finite Field", 1994 Information Theory and Its Applications, pp.239-244, 1994

[8] E. Okamoto, T. Uematsu and M. Mambo: "Permutation Cipher scheme Using polynomials over a Field", Transactions of the IEICE(The Institute of Electronics, Information and Communication Engineers), vol.E78-D, no.2, pp.138-142, 1995

[9] G. Solomon: "Optimal frequency Hopping Sequences for Multiple Access", Proc. of the Symp. on Spread Spectrum Commun., pp.33-35, 1973

[10] J. H. van Lint and R. M. Wilson: "A Course in Combinatorics", Cambridge University Press, 1992

[11] Y. Tamura and E. Okamoto: "Concept and Implementation of Flexible Secret Sharing Scheme", Proc. of CSS'98, 1998

[12] K. Martin and J. Nakahara, Jr: "Updating the Parameters of an Established Threshold Scheme", Private Communications, 1998

[13] Y. Tamura, M. Tada and E. Okamoto: "Update of Access Structure in Shamir's (k, n) Threshold Scheme", Proc. of SCIS'99, 1999

社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
ISEC2003-32 (2003-07)

"just" k-out-of-n 署名方式の提案

岡本　健[†]　岡本　栄司[†]

† 筑波大学 電子・情報工学系　〒305-8573 茨城県つくば市天王台 1-1-1
E-mail: †{ken,okamoto}@is.tsukuba.ac.jp

あらまし　本稿では，新しい k-out-of-n 署名を提案する．従来方式の場合，検証者は n 人のグループのうち少なくとも (at least) k 人がメッセージに同意していることは確認できるが，実際に k 人かどうかはわからなかった．これに対して提案方式の場合，検証者は過不足なく (just) k 人が同意していることを確認できる．提案方式では，このようなシステム実現のために共通鍵暗号系に基づく関数を明示的に利用している．本稿では，最初にこの関数をどのようにして署名技術に利用するかについて説明する．次に既存方式の 1-out-of-n 署名にこの技術を適用することにより，"at least" および "just" のいずれの方式にも拡張可能であることを示す．
キーワード　電子署名, k-out-of-n 署名, ring 署名, 匿名性, 共通鍵暗号系

Proposal of "just" k-out-of-n signatures

Takeshi OKAMOTO[†] and Eiji OKAMOTO[†]

† Institute of Information Sciences and Electronics, University of Tsukuba　1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
E-mail: †{ken,okamoto}@is.tsukuba.ac.jp

Abstract　In this paper, we propose new k-out-of-n signature schemes. As far as the knowledge of authors, all the previous schemes are "at least" k-out-of-n signatures. This means that in the signature systems, at least k persons agree on the message to be signed. More concretely, we propose "just" k-out-of-n signature schemes. To achieve such features, we explicitly use symmetric-key cryptosystems in our proposed scheme. As for the paper's outline, we first show the basic idea of our signature schemes. Later we explain how to use this cryptosystem. Finally, we propose both "at least" and "just" schemes by using previous 1-out-of-n schemes.
Key words　digital signature, k-out-of-n signature, ring signature, anonymous signer, symmetric-key cryptosystem

1. は じ め に

k-out-of-n 署名(注1) は n 人のグループのうち，k 人の秘密情報を用いることによって，署名を作成する方式である．署名の検証時において，検証者は n 人のうちの k 人がメッセージに対して同意していることを確認できるが，その k 人は誰であるかはわからない．このような特徴をもつ署名技術は，内部告発者の保護に有益であり，実社会において今後益々求められる技術である．

また，1-out-of-n 署名は，k-out-of-n 署名の特殊な形式であると考えることができ，グループ内の 1 人のメンバによって署名が作成できるため，内部告発に重点を置いた方式といえる．これに対し，k-out-of-n 署名は内部告発が可能であるという機能に加え，署名の検証により「グループ内の何人がメッセージに同意したか」という情報についても得ることができる．また状況によっては，議員や役員の罷免というように，メッセージの内容というよりも，何人が同意しているかということに重点を置く場合も考えられる．

これまでの研究については，1-out-of-n 署名の場合，Ring 署名 [5] と呼ばれる，落とし戸付き一方向性置換を用いた方式が Rivest らによって提案されている．離散対数問題に基づく方式については，Cramer らによる方式がある [3]．大久保らは上記の方式よりも署名長の削減が可能な方式を提案した [1]．k-out-of-n については，Bresson ら [2] や菊池ら [7] による方式があり，これらはいずれも Ring 署名を拡張した方式である．また，通常の k-out-of-n に否認付加機能等，種々の機能を付加した方式についてもこれまでに数多く発表されている [4] [8] [9]．

ここで注意すべきことは，従来の k-out-of-n 署名はいずれも，少なくとも (at least) k 人のメンバが集まれば，署名が作

(注1)：k-out-of-n signature は，k-out-of-n 署名と k-out-of-n 証明の 2 種類の呼び方があるが，本稿では前者を採用する．

成できるという点である．例えば，n 人以下の $k+\alpha$ 人が秘密情報を利用して署名を作成しても，k-out-of-n 署名を作成することは可能である．このことは，「k 人の同意を得た」という情報についてあいまいさを残すことになり，実社会の要求にうまく適応できない場合が考えられる．

このような問題点を解決するために，本稿では検証によって得られる情報の達成度に応じて，k-out-of-n 署名を "at least" 方式と "just" の 2 種類に分類する．前者は既に述べたように，少なくとも k 人，すなわち k 人以上が同意することにより署名を作成する方式であり，後者は過不足なく (just) k 人が同意することにより署名を作成する方式である．

例として，ある役員を罷免するという内容の just k-out-of-n 署名が会議に提出された場合を考える．このとき，会議の運営者側は k 人については罷免に同意しているが，$n-k$ 人については同意していない，あるいは署名作成に関与していないという情報を得ることができる．このことは at least 方式より多くの情報を得ることができるため，この問題に対応するための有益な判断材料になりうる．

本稿では離散対数問題に基づく新しい k-out-of-n を提案する．提案する署名技術には共通鍵系に基づく暗号関数が明示的に含まれており，この技術を [1] や [3] の方式に適用することにより，既存方式の 1-out-of-n から k-out-of-n の署名方式に拡張することができる．また用いる共通鍵暗号関数の個数を変えることにより，at least 方式，あるいは，just 方式のいずれにも拡張できる．

既存方式を at least 方式と just 方式に拡張し，両者を比較した場合，署名生成や検証にかかる計算量コストは同程度である．しかしながら，伝送量に関しては，後者の方がわずかに大きい，例えば公開鍵となる法 p のサイズが 1024 ビットの場合，後者の方が，1024 k ビット程度大きくなる．また，セキュリティーパラメータが同じ場合，公開鍵のサイズは両者とも同程度である．

安全性については，利用する共通鍵暗号関数がランダム関数であるというように，理想的な関数を仮定した場合，オリジナルの方式と同程度になる．また，計算量については，オリジナル方式と比較した場合，k 倍程度大きくなる．これは，オリジナル方式が 1-out-of-n 署名であるのに対して，提案方式は k-out-of-n 署名であり，提案方式は k ラウンドの計算処理を行うためである．

本稿では，最初に 2. 章で提案方式に関する本質的なアイデアを説明する．3. 章では，提案する署名技術の理解を助けるために，グループが 1 人の場合の単署名について述べる．この署名技術を既存の方式に適用することによって，4. 章では at least k-out-of-n 署名，5. 章では just k-out-of-n 署名がそれぞれ実現できることを示す．6. 章で提案方式に対する安全性，効率性について考察した後，7. 章でまとめる．

2. 提案方式のアイデア

提案方式では共通鍵暗号関数を明示的に利用している．本章ではこの関数がどのように機能しているかについて説明する．

2.1 準　　備

関数 E： E は $E:\mathbb{Z}_p \to \mathbb{Z}_p$ (p は素数) を満たす共通鍵暗号の暗号化関数とする．E^{ℓ} は ℓ 回の暗号化処理を表す．反対に $E^{-\ell}$ は ℓ 回の復号化処理を表す．ここで，$x \in \mathbb{Z}_p$ に対して，$E^{-\ell}(E^{\ell}(x)) = x$, $E^0(x) = E^{-0}(x) = x$ となることに注意．また簡単化のため，$E^1_K = E_K$ とする．

2.2 共通鍵暗号関数を用いた試行

提案手法のアイデアを説明するために，以下のような手順を試みる．ここでは，ユーザ A, B が存在し，A は問題作成者，B は回答者となる．

パラメータ生成： p, q は $q|p-1$ を満たす 2 つの大きな素数とする．g は p における位数 q の部分群の生成源とする．整数 ℓ を生成する．乱数 $s \in \mathbb{Z}_\ell$ を生成する．

パラメータ管理： (p, q, g) および (E, ℓ) は公開情報とする．s は A の秘密情報とする．

問題作成： A は以下を実行する．
Step1　乱数 $r \in \mathbb{Z}_q$ を生成し，$x = g^r \bmod p$ を計算する．
Step2　$Q_0 = E^{-s}(x)$ を計算する．
Step3　Q_0 を B に送信する．

回答： B は以下を実行する．
Step1　$i = 0, 1, \cdot, \ell - 1$ に対して，ℓ 個のパラメータ $Q_{(i)} = E^i_K(Q_i)$ を計算する．
Step2　B は何らかの計算により，$s' \in \mathbb{Z}_\ell$ を選択する．
Step3　s' を A に送信する．

解の開示： A, B は以下を実行する．
Step1　A は (r, s) を B に送る．
Step2　B は受け取った (r, s) を用いて，$g^r = E^s(x) \bmod p$ が成り立つか確認する．成り立たなければプロトコルを停止する．
Step3　B は $s = s'$ が成り立つか確認する．成り立つならば，B の勝ちとする．そうでなければ A の勝ちとする．

2.3 考　　察

E がランダム関数というように理想的な関数と仮定する．このとき，以下のことがいえる．

(1)　A が不正を行う，すなわち A が $g^{s_1} = E^{a_1}(Q_0)$, $g^{s_2} = E^{a_2}(Q_0)$ $(a_1 \neq a_2, a_1, a_2 \in \mathbb{Z}_\ell)$ となるような値 s_1, s_2 を用意することにより，A の方へ勝ちを誘導する事は，離散対数問題を解くのと同程度に難しい。

(2)　B がどのような計算手法を用いても B が正解する確率は、$1/\ell$ である．

(1) について： A は Q_s の構造については $x = E^s(Q_0) = Q_s = g^r \bmod p$ であるということを知っている．しかしなが

ら，その他のパラメータ，例えば $E^{s+1}(Q_0)$ についてはわからない．これは，Q_s 以外のパラメータがが暗号化処理されているため，構造が崩されているためである．提案方式では，「あるユーザが指定したパラメータ以外は，構造が崩れる」という性質を利用して，提案方式の目的，すなわち "at least" 方式あるいは "just" 方式を実現している．

(2) について： パラメータ Q_0 は，$Q_0 = E^{-s}(Q_s)$ というように s 回の復号化処理によって得られる．しかしこの処理過程を知らないユーザは，Q_0 から何回の暗号化処理を行えば Q_s になるかわからない．もし E がランダム関数であるならば，$Q_0, Q_1, \cdots, Q_{\ell-1}$ のそれぞれが Q_s になる確率は，すべて同じである．提案方式の安全性は，このような安全性に基づいており，n 人のグループの中から k の署名者を特定することは，情報理論的に困難である．

3. 提案方式の特徴

本章では提案する署名方式について説明する．提案方式は Schnorr 署名 [6] を拡張した方式であるが，共通鍵暗号関数を明示的に付加しているという点でオリジナル方式と異なる．簡単化のために，1-out-of-1 (すなわち、通常の署名) で構成された場合の手順を以下に示す．

鍵生成: p, q は $q|p-1$ を満たす 2 つの大きな素数とする．g は p における位数 q の部分群の生成源とする．s, v は $v = g^{-s} \bmod p$ を満たす整数とする．$\mathcal{H}$ は $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ を満たすハッシュ関数とする．

公開鍵/秘密鍵: 署名者の公開鍵を (p, g, v)，秘密鍵を s とする．

署名生成: 署名者は以下を実行する．

Step1 乱数 $r \in \mathbb{Z}_q$ を生成し，$x = g^r \bmod p$ を計算する．

Step2 乱数 $\alpha \in \mathbb{Z}_q$ を生成し，$T = g^\alpha \bmod p$ を計算する．

Step3 $c = \mathcal{H}(x||T||m)$ を計算する．

Step4 $y = r + s - \alpha c \bmod q$ を計算する．

署名: メッセージ m に対する署名を (c, y, T) とする．

検証: 検証者は以下を実行する．

Step1 $x = g^y T^c v \bmod p$ を計算する．

Step2 $c' = \mathcal{H}(x||T||m)$ を計算する．

Step3 $c = c'$ が成り立つかどうか確認する．成り立つならば受理する．そうでなければ，受理しない．

提案方式の主な特徴は，

(1) $v = g^{-s} \bmod p$ となる $s \in \mathbb{Z}$ を知っている．

(2) $T = g^\alpha \bmod p$ となる $\alpha \in \mathbb{Z}$ を知っている．

という 2 つの条件を満たしたユーザに限り署名が作成できるという点である．提案方式では，上記の署名生成における Step1 から Step4 までの手順を 1 ラウンドとし，k-out-of-n ならば k ラウンド分の計算処理を行う．また各ラウンドに対して，k 人のメンバの中の 1 人が担当し，そのメンバが該当するラウンドの署名 (の一部) を作成する．こうして k 人分の署名が集まったとき k-out-of-n 署名方式の署名が作成されたことになる．

本章で示した提案方式は，[1] あるいは [3] の方式を適応して，k-out-of-n 署名を実現できる．本稿では [1] の方式を適応し，具体的な手順を示す．

4. "at least" k-out-of-n 署名

本章では，前章で示した署名技術を [1] に適用する．これにより，at least 1-out-of-n から at least k-out-of-n に拡張できることを示す．

4.1 準　　備

集合 $\mathcal{S}$： $\mathcal{S}$ は署名に参加するユーザの集合とする．

関数 f： f は $\mathcal{S}$ に属するユーザが何ラウンド目に署名計算を行うかを求める関数とする．例えば $i \in \mathcal{S}$ のとき $f(i) = \gamma$ は，i は γ ラウンド目に計算処理を行うことを意味する．反対に f^{-1} は f の逆関数であり，$f^{-1}(\gamma) = i$ となる．ユーザ番号とラウンド数の関係は，

(1) 任意の $i, i' \in \mathcal{S}$ に対して，$i < i'$ ならば $f(i) < f(i')$．

(2) ラウンドは $0, 1, \cdots, k-1$ と進む．

となる．例えば，2-out-of-5 署名で，$\mathcal{S} = \{0, 2\}$ の場合，$f(0) = 0$, $f(2) = 1$, $f^{-1}(0) = 0$, $f^{-1}(1) = 2$, となる．

関数 E_i： E_i $(i = 0, 1, 2, \cdots, n-1)$ は $E_i : \mathbb{Z}_{q_i} \to \mathbb{Z}_{q_i}$ を満たす共通鍵暗号関数とする．

関数 $\mathcal{H}_i$： $\mathcal{H}_i$ $(i = 0, 1, 2, \cdots, n-1)$ は $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_{q_{i+1}}$: を満たすハッシュ関数とする．

パラメータ $\alpha_i^{(j)}$： パラメータ α に対して，$\alpha_i^{(j)}$ という表記は，j ラウンド目のユーザ i のパラメータであることを示す．

4.2 提 案 方 式

鍵生成: 3 章で記述された手順により，$(p_i, q_i, g_i, v_i, s_i)$ を生成する．

公開鍵/秘密鍵: ユーザ i $(0, 1, 2, \cdots, n-1)$ の公開鍵を (p_i, g_i, v_i)，秘密鍵を s_i とする．

署名生成: $i \in \mathcal{S}$ を満たす各ユーザ i は，以下を実行する．

Step1 (共通鍵暗号パラメータの初期化)

```
% アルゴリズム 4-1
for ($i = 0, 1, \cdots, n-1$) do
{
  if ($i \in \mathcal{S}$) do
  {
    $\gamma := f(i)$. % ラウンドの決定
    ユーザ $i$ は以下を実行:
      乱数 $\alpha_i \in \mathbb{Z}_q$ を生成する.
```

```
        $T_i^{(\gamma)} = g_i^{\alpha_i} \bmod p_i$ を計算する.
        $T_i^{(0)} = E_i^{-\gamma}(T_i^{(\gamma)})$ を計算する.
    }

    else do
    {
      $u \in \mathcal{S}$ を満たすユーザ $u$ (誰でもよい) は
                          乱数 $T_i^{(0)} \in \mathbb{Z}_{q_i}$ を生成する.
    }

}

Output $(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$.
```

Step2 (k ラウンドの署名生成)

$$T = T_0^{(0)} \| T_1^{(0)} \| \cdots \| T_{n-1}^{(0)}$$

とする.

```
% アルゴリズム 4-2

for ($j = 0, 1, \cdots, k-1$) do
{
  $i := f^{-1}(j)$. % ユーザの決定

  ユーザ $i$ は以下を実行する:
      乱数 $r_i \in \mathbb{Z}_{q_i}$ を生成する.
      $x_i = g_i^{r_i} \bmod p_i$ を計算する.
      $c_{i+1} = \mathcal{H}_i(x_i \| T \| m)$ を計算する.

  for  ($t = i+1, \cdots, n-1, 0, 1, \cdots, i-1$) do
  {
    ユーザ $i$ は以下を実行する:
        乱数 $y_t \in \mathbb{Z}_{q_t}$ を生成する.
        $T_t = E_t^j(T_t^{(0)})$ を計算する.
        $x_t = g_t^{y_t} T_t^{c_t} \bmod p_t$ を計算する.

    if ($i \neq n-1$) do
    {
      $c_{i+1} = \mathcal{H}_i(x_i \| T \| m)$ を計算する.
    }

    else do
    {
      $c_0 = \mathcal{H}_i(x_i \| T \| m)$ を計算する.
    }
```

```
  }

  ユーザ $i$ は $y_i = r_i + s_i - \alpha_i c_i$ を計算する.

  $x_0^{(j)} := x_0$,
  $y_0^{(j)} := y_0, y_0^{(j)} := y_0, \cdots, y_{n-1}^{(j)} := y_{n-1}$.
}

Output  $(x_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)})$,
        $(x_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)})$,
                          ⋮
        $(x_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)})$.
```

署名: メッセージ m に対する署名を

$$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)}),$$
$$(c_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)}),$$
$$(c_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)}),$$
$$\vdots$$
$$(c_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)}).$$

とする.

検証: 検証者は以下を実行する.

```
% アルゴリズム 4-3

for ($j = 0, 1, \cdots, k-1$) do
{

  $c_0 := c_0^{(j)}, y_0 := y_0^{(j)}, \cdots, y_{n-1} := y_{n-1}^{(j)}$.

  for ($i = 0, 1, \cdots, n-1$) do
  {
    $x_i = g_i^{y_i} T_i^{c_i} v_i \bmod p_i$ を計算する.
    $T_i = E_i^j(T_i^{(0)})$ を計算する.
    $c_{i+1} = \mathcal{H}_i(x_i \| T \| m)$ を計算する.

  }

  等式 $c_0 = c_n$ が成り立つかどうか確認する.
}

if (すべての等式が成り立つ) do
{
  Output "受理する".
}
else do
{
  Output "受理しない".
}
```

5. “just” k-out-of-n 署名

本章では前章で示した方式に対し，さらに共通鍵暗号関数を付加することによって，"just" k-out-of-n 方式が実現できることを示す.

5.1 準　　備

関数 $\bar{E}_i$:　$\bar{E}_i$ $(i = 1, 2, \cdots, n-1)$ は $\bar{E}_i : \mathbb{Z}_a \to \mathbb{Z}_a$ (a は整数) を満たす共通鍵暗号関数とする.

5.2 提 案 方 式

鍵生成: 3 章で記述された手順により，$(p_i, q_i, g_i, v_i, s_i)$ を生成する.

公開鍵/秘密鍵: ユーザ i の公開鍵を (p_i, q_i, g_i, v_i)，秘密鍵を s_i とする.

署名生成: $i \in S$ を満たす各ユーザ i は，以下を実行する.

Step1 (共通鍵暗号パラメータの初期化: T の生成) 4 章で記述された手順により，$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$ を生成する..

Step2 (共通鍵暗号パラメータの初期化: $\bar{T}$ の生成)

```
% アルゴリズム 5-1

for ($j = 0, 1, \cdots, k-1$) do
{
    $i := f^{-1}(j)$. % ユーザの決定

    ユーザ $i$ は以下を実行:
        乱数 $\tilde{\alpha}_i \in \mathbb{Z}_{q_i}$ を生成する.
        $\tilde{T}_i^{(\gamma)} = g_i^{\tilde{\alpha}_i} \bmod p_i$ を計算する.
        $\tilde{T}_0^{(\gamma)} = \bar{E}^{-i}\ (\tilde{T}_i^{(\gamma)})$ を計算する.
}

Output $(T_0^{(0)}, T_0^{(1)}, \cdots, T_0^{(k-1)})$.
```

Step3 (k ラウンドの署名生成)

$$T = T_0^{(0)} \| T_1^{(0)} \| \cdots \| T_{n-1}^{(0)} \| T_0^{(0)} \| T_0^{(1)} \| \cdots \| T_0^{(k-1)}$$

とする.

```
% アルゴリズム 5-2

for ($j = 0, 1, \cdots, k-1$) do
{
    $i := f^{-1}(j)$. % ユーザの決定

    ユーザ $i$ は以下を実行する:
        乱数 $r_i \in \mathbb{Z}_{q_i}$ を生成する.
        $x_i = g_i^{r_i} \bmod p_i$ を計算する.
        $c_{i+1} = \mathcal{H}_i(x_i \| T \| m)$ を計算する.

    for ($t = i+1, \cdots, n-1, 0, 1, \cdots, i-1$) do
    {
        ユーザ $i$ は以下を実行する:
            乱数 $y_t \in \mathbb{Z}_{q_t}$ を生成する.
            $T_t = \bar{E}_t^j(T_t^{(0)})$ を計算する.
            $\bar{T}_t = E^t(T_0^{(j)})$ を計算する.
            $x_t = g_t^{y_t}(T_t \bar{T}_t)^{c_t} \bmod p_t$ を計算する.

        if ($i \neq n-1$) do
        {
                $c_{t+1} = \mathcal{H}_t(x_t \| T \| m)$ を計算する.
        }

        else do
        {
                $c_0 = \mathcal{H}_t(x_t \| T \| m)$ を計算する.
        }

    }

    ユーザ $i$ は $y_i = r_i + s_i - c_i(\alpha_i \tilde{\alpha}_i)$ を計算する.

    $x_0^{(j)} := x_0$,
    $y_0^{(j)} := y_0, y_0^{(j)} := y_0, \cdots, y_{n-1}^{(j)} := y_{n-1}$.
}

Output  $(x_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)})$,
        $(x_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)})$,
                        ⋮
        $(x_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)})$.
```

署名: メッセージ m に対する署名を

$$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)}),$$
$$(T_0^{(0)} \| T_0^{(1)} \| \cdots \| T_0^{(k-1)})$$
$$(c_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)}),$$
$$(c_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)}),$$
$$\vdots$$
$$(c_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)}).$$

とする.

表1 性能評価
Table 1 Performance evaluation.

方式	署名の分類	共通鍵暗号関数の個数	署名長
オリジナル方式	at least 1-out-of-n 署名	なし	$\|c_0\| + \|y_i\|$ $(0 \leq i < n)$
Scheme I	at least k-out-of-n 署名	n	$\|T_i^{(0)}\| + \|c_0^{(j)}\| + \|y_i^{(j)}\|$ $(0 \leq i < n), (0 \leq j < k)$
Scheme II	just k-out-of-n 署名	$n+k$	$\|T_i^{(0)}\| + \|\bar{T}_0^{(j)}\| + \|c_0^{(j)}\| + \|y_i^{(j)}\|$ $(0 \leq i < n), (0 \leq j < k)$

検証: 検証者は以下を実行する.

```
% アルゴリズム 5-3
for ($j = 0, 1, \cdots, k-1$) do
{
  $c_0 := c_0^{(j)}, y_0 := y_0^{(j)}, \cdots, y_{n-1} := y_{n-1}^{(j)}$.

  for ($i = 0, 1, \cdots, n-1$) do
  {
    $T_i = E_i^j(T_i^{(0)})$ を計算する.
    $\tilde{T}_i = \tilde{E}^i(T_0^{(j)})$ を計算する.
    $x_i = g_i^{y_i}(T_i\tilde{T}_i)^{c_i} v_i \bmod p_i$ を計算する.
    $c_{i+1} = \mathcal{H}_i(x_i||T||m)$ を計算する.
  }
  等式 $c_0 = c_n$ が成り立つかどうか確認する.
}
if (すべての等式が成り立つ) do
{
  Output "受理する".
}
else do
{
  Output "受理しない".
}
```

6. 考　察

本章では，提案方式についての安全性や性能評価について考察する.

[1] の離散対数問題に基づく方式をオリジナル方式，4章，5章で提案した方式をそれぞれ，Scheme I, Scheme II とし，両者の性能評価を表1に示す.

署名長については，提案方式の場合，オリジナル方式に比べ，(1) 共通鍵暗号関数によって生成された値，(2) k ラウンドの計算処理によって生成された値，の分だけ増加する．また，提案方式の両者を比較した場合，Scheme I に比べ Scheme II の方が共通鍵暗号関数の個数が多いので，これによって生成された分だけ署名長が長くなる.

より具体的な値として，各ユーザの鍵を $|P_i| = 512, |q_i| = 160$ $(0 \leq i < n)$ としたとき，オリジナル方式の署名は，$672n + 160$ ビット程度，Scheme I, Scheme II はそれぞれ，$672kn + 1024n + 160k$ ビット，$672kn + 1024n + 2624k$ ビット程度になる.

安全性については，共通鍵暗号関数が理想的な関数であると仮定した場合，提案方式はいずれも，オリジナル方式と同程度である.

計算量については，提案方式とオリジナル方式と比べた場合，k 倍程度，計算量が増加する．これは，提案方式が k ラウンドの計算処理を行うためである．提案方式の両者を比較した場合，計算量は同程度である．Scheme I に比べ Scheme II の方が，より多く共通鍵暗号関数を用いた暗号処理を行っている．しかし，これは公開鍵暗号関数の計算処理と比べれば，大きな負荷にはならない.

7. ま と め

本稿では共通鍵暗号系に基づく関数を用いることによって，既存方式の 1-out-of-n 署名を k-out-of-n 署名に拡張できることを示した．また k-out-of-n 署名を at least 方式と just 方式に分類し，従来方式では at least 方式しか実現できなかったが，提案した署名技術を用いることによって，just 方式が実現できることを示した.

文　献

[1] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *Asiacrypt '02*, LNCS No. 2501, pages 415–432. Springer-Verlag, 2002.
[2] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *Crypto '02*, LNCS No. 2442, pages 465–480. Springer-Verlag, 2002.
[3] R. Cramer, I. Bjerre Damgard, and B. Schoenmakers. Proof of partial knowledge and simplified design of witness hiding protocols. In *Crypto '94*, LNCS No. 839, pages 174–187. Springer-Verlag, 1994.
[4] Moni Naor. Deniable ring authentication. In *Crypto '02*, LNCS No. 2442, pages 481–498. Springer-Verlag, 2002.
[5] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Asiacrypt '01*, LNCS No. 2248, pages 552–565. Springer-Verlag, 2001.
[6] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.
[7] 菊池浩明, 多田美奈子, 中西祥八朗. Ring signature に基づいた k-out-of-n 証明の提案. コンピュータ セキュリティ シンポジウム (CSS'99), pages 83–87, 2002.
[8] 菊池明, 多田美奈子, 中西祥八朗. リング署名プロトコルにおける署名者開示. 情報処理学会コンピュータセキュリティ研究発表会 (CSEC-20-27)), pages 149–153, 2003.
[9] 須賀祐治, 岩村恵市. 否認機能を持つリング署名方式. 暗号と情報セキュリティシンポジウム (SCIS'03), pages 435–439, 2003.
[10] 大久保美也子, 阿部 正幸, 鈴木幸太郎, 辻井重男. 署名長が短い 1-out-of-n 証明. 暗号と情報セキュリティシンポジウム (SCIS'02), pages 189–193, 2002.

SCIS 2004 The 2004 Symposium on Cryptography and Information Security
Sendai, Japan, Jan.27-30, 2004
The Institute of Electronics, Information and Communication Engineers

k-out-of-n署名の拡張
Extended k-out-of-n Signature

樋口達巳* Tatsumi HIGUCHI　　岡本　健 † Takeshi OKAMOTO　　岡本　栄司 † Eiji OKAMOTO

あらまし 本稿では, k-out-of-n 署名の拡張を提案する. 従来研究では, グループに加入している n 人のうち, 少なくとも k 人が参加していることが確認できる k-out-of-n 署名や, ちょうど (just) k 人の参加が確認できる "just"k-out-of-n 署名が提案されている. 我々はこれらの方式を利用して, 署名に参加しているユーザ数をより詳細に設定できる方法を試みる. 本提案方式では, 署名者が l 人以上, m 人以下である署名を作成することができ, またこのような署名技術は, 内部告発者により効果的な保護を与えることができる.

キーワード リング署名, 1-out-of-n 署名, k-out-of-n 署名, 匿名性, 共通鍵暗号系

1 はじめに

1-out-of-n 署名は n 人のうちの少なくとも 1 人が署名を作成する方式であり, 検証者側は n 人のうちの誰が署名を作成したか知ることができない.

また k-out-of-n 署名は n 人のグループのうちの, k 人の秘密情報を用いることによって署名を作成する方式である. 署名の検証時には, 1-out-of-n 署名と同様にグループのユーザ n 人のうちの, k 人が秘密情報を用いていることがわかるが, そのk人が誰であるかはわからない. このような特徴を持つ署名技術は, 内部告発者の保護に利用できる.

これまでの研究では, Rivest らによって落とし戸付き一方向性置換を用いた Ring 署名 [2] が, Cramer らによって離散対数問題に基づいた方式 [3] が提案されている. また [3] は大久保らによって証明長の短い方式 [4] が提案された. さらに [2] や [4] の 1-out-of-n 署名は Bresson ら [5] や菊池ら [6] によって k-out-of-n 署名に拡張されている. Ring 署名や 1-out-of-n 署名は様々な機能を付加された方式が提案されており, Ring 署名に否認機能を付加した方式 [8] や, 署名者の開示機能を付加した方式 [7] もある.

さらに k-out-of-n 署名を少なくとも (at least) k なのか, ちょうど (just) k なのかを, 明示的に使い分けることができる方式が岡本ら [1] によって提案されている. [1] は離散対数に基づく k-out-of-n 署名で, [3], [4] の方式に共通鍵系に基づく暗号関数を用いることによって構成できる. [1] はちょうど k 人の同意が必要である様な実社会の要求に応えることができ, 少なくとも (at least) k 人というあいまいさを消すことができる.

本稿では [1] の "just"k-out-of-n 署名を拡張して, 署名者数をより詳細に設定できる方式を提案する. 例えば実際に署名作成に協力したユーザが 5 人であるのに対し, 3 人～7 人のユーザの署名, といった様に署名者数に幅を持たせた署名が作成できる. 検証者は署名を受理したなら, 3 人～7 人のユーザが署名作成に協力していることがわかる. 実際の署名者数が少なかったとしても, 多くの署名者がいるように設定できるので, この方式は多様な内部告発モデルに応用できると考えられる. 実際, 企業や組織においての内部告発は, 内部告発者にとって多大なリスクを負うものである. 1-out-of-n 署名によって内部告発者の秘密は保障されているものの, 内部告発者にとっての安心材料は多いに越したことはないはずである. 提案方式を用いることによって, 署名者は内部告発によって被る不利益の不安が減るように, 署名者数の範囲を設定することができる.

本稿の構成としては, 2 章で提案方式の元となる just k-out-of-n 署名のプロトコルを紹介し, 3 章で提案する署名プロトコルを示す. 4 章においては安全性, 署名長などの考察をし, 5 章でまとめる.

* 筑波大学第三学群情報学類 〒 305-8573 茨城県つくば市天王台 1-1-1, College of Information Sciences, Third Cluster of Colleges, University of Tsukuba, 1-1-1 Tennoudai Tsukuba, Ibaraki, 305-8573, Japan
† 筑波大学電子・情報工学系 〒 305-8573 茨城県つくば市天王台 1-1-1, Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennoudai Tsukuba, Ibaraki, 305-8573, Japan

2　従来方式

2.1　1-out-of-n 署名

ここでは大久保らによって提案された方式 [4] を説明する．これは離散対数問題に基づく証明長の短い 1-out-of-n 証明方式である．

2.1.1　準備

p, q を $q \mid q-1$ を満たす大きな素数，g を p の位数 q の部分群の生成元とし，(p, q, g) を公開する．署名生成ユーザ i は $x_i \in Z_q$ を秘密鍵とし，$y_i = g^{x_i} \bmod p$ を公開鍵とする．また一方向性セキュアハッシュ関数を $H()$ とする．

2.1.2　署名作成

step1 署名作成ユーザ i は以下を求める

$$T_i := g^{\alpha} \bmod p$$
$$c_{i+1} := H(m \| T_i)$$
ただし$\alpha \in_U Z_q$とする．

step2 $(j = i+1, \cdots, n, 1, \cdots, i-1)$ について以下を順次計算する．

$$T_j := g^{s_j} y_j^{c_j} \bmod p$$
$$c_{j+1} := H(m \| T_j)$$
ただし $s_j \in_U Z_q$とする

2.1.3　署名検証

$(j = 1, 2, \cdots, n)$ について，以下を繰り返す．

$$T_j := g^{s_j} y_j^{c_j} \bmod p$$
$$c_{j+1} := H(m \| T_j)$$

$c_1 = c_{n+1}$ならば署名を受理し，
そうでなければ受理しない．

2.2　"at least"k-out-of-n 署名

本節では [1] の方式の概要を説明する．これは Schnorr 署名を拡張した方式である．署名者はリングを k 個作成し，あるユーザが秘密鍵を用いてリングを閉じる処理 1 回を 1 ラウンドとすると，k 回のラウンドで k 個のリングを全て閉じる．この k ラウンドにおいて，共通鍵系に基づく暗号関数を用いることによって，1 人のユーザは 1 回だけしかリングを閉じることのできない仕組みになっている．この仕組みによって少なくとも k 人の署名者がいることがわかる．

2.2.1　準備

パラメータ	定義
集合 S	署名に参加するユーザの集合
$f()$	$f()$ は S に属するユーザが何ラウンド目に署名計算を行うか求める関数
$E_i()$	$E_i()(i = 0, 1, \cdots, n-1)$ は $\mathbb{Z}_{q_i} \rightarrow \mathbb{Z}_{q_i}$ を満たす共通鍵暗号関数， E^lは l 回の暗号化処理， E^{-l}は l 回 の復号化処理， また簡単化のため $E^1 = E$ とする．
H_i	H_i: $(0,1)^* \rightarrow \mathbb{Z}_{q_{i+1}}$ を満たす 一方向性セキュアハッシュ関数
(p_i, q_i, g_i, v_i)	ユーザ i の公開鍵
s_i	ユーザ i の秘密鍵
$\alpha_i^{(j)}$	j ラウンド目のユーザ i のパラメータ α

2.2.2　署名作成

本節では署名作成に必要な前処理とその概要を示す．詳細に関しては [1] を参照．

step1(共通鍵暗号パラメータの初期化: T の生成)
$(i = 0, 1, \cdots, n-1)$ について，以下を繰り返す．

```
if($i \in S$)
    $\gamma := f(i)$    %ラウンドの決定

    署名作成ユーザ $i$ は
        以下を順次計算する．
    $T_i^{(\gamma)} = g_i^{\alpha_i} \bmod p_i$
    $T_i^{(0)} = E_i^{-\gamma}(T_i^{(\gamma)})$
    ただし$\alpha_i \in Z_q$とする
else
    $u \in S$ を満たすユーザ $u$ は
        乱数 $T_i^{(0)} \in Z_{q_i}$を生成
```

step1 の結果として $(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$ を出力する．

step1 で求めた $(T_i^{(0)})$(以下 $i = 0, 1, \cdots, n-1$) から $E_i()$ によって，$(T_i^{(j)})(j = 0, 1, \cdots, n-1)$ を求めることができる．しかし各ラウンドでリングを閉じることができるのは，$T_i = g^{\alpha_i} \bmod p_i$ の α_i を知っているユーザ i が s_i を用いることによってのみである．

2.2.3　署名出力

メッセージ m に対する署名として
$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$,
$(c_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)})$,

$(c_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)})$,

$\vdots$

$(c_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)})$
を出力する.

2.2.4 署名検証

$(j = 0, 1, \cdots, n-1)$ について, 以下を繰り返す.

```
$c_0 := c_0^{(j)}, y_0 := y_0^{(j)}, \cdots, y_{n-1} := y_{n-1}^{(j)}$

$(i = 0, 1, \cdots, n-1)$ について,
以下を順次計算する.
    $T_i = E_i^j(T_i^{(0)})$
    $x_i = g_i^{y_i} T_i^{c_i} v_i \bmod p_i$
    $c_{i+1} = H_i(x_i \| T \| m)$
等式 $c_0 = c_n$ が成り立つか確認する
```

全ての等式 $c_0 = c_n$ が成り立てば署名を受理し, 成り立たなければ受理しない.

2.3 "just"k-out-of-n 署名

本節では提案方式の基礎となる [1] の "just"k-out-of-n 署名を概説する. 前節の "at least"k-out-of-n 署名はラウンド間に共通鍵暗号関数を用いて, 1人のユーザが1回だけしかリングを閉じれないように工夫をしていた. さらに今回は各ラウンドのユーザ間にも共通鍵暗号関数を用いることによって, 各ラウンドでは1人のユーザしかリングを閉じることができないようになっている. 従ってkラウンドの処理によって全てのリングが閉じているならば, ちょうどk人がリングを閉じるために秘密情報を用いたことを意味する.

2.3.1 準備

パラメータ	定義
$\tilde{E}_i$	$\tilde{E}_i : \mathbb{Z}_a \to \mathbb{Z}_a$($a$ は整数) を満たす共通鍵暗号関数
(p_i, q_i, g_i, v_i)	ユーザ i の公開鍵
s_i	ユーザ i の秘密鍵

そのほかのパラメータは 2.2 章と同様とする.

2.3.2 署名作成

本項では前節の署名作成段階と同様に, 署名作成においての前処理と概要を説明する. 署名作成プロトコルの詳細に関しては [1] を参照.

step1 (共通鍵暗号パラメータ T の初期化)
2.2.2 章の署名作成 step1 と同様にして,
$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$ を得る.

step2 (共通鍵暗号パラメータ $\tilde{T}$ の初期化)
$(j = 0, 1, \cdots, k-1)$ について, 以下を繰り返す.

```
$i := f^{-1}(j)$    %ユーザの決定

署名作成ユーザ $i$ は
        以下を順次計算する.
$\tilde{T}_i^{(\gamma)} = g_i^{\tilde{\alpha}_i} \bmod p_i$
$\tilde{T}_0^{(\gamma)} = \tilde{E}^{-i}(\tilde{T}_i^{(\gamma)})$
ただし$\tilde{\alpha}_i \in Z_{q_i}$とする
```

step2 の結果として $(\tilde{T}_0^{(0)}, \tilde{T}_0^{(1)}, \cdots, \tilde{T}_0^{(k-1)})$ を出力する.

2.3.3 署名出力

メッセージ m に対する署名として
$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$,
$(\tilde{T}_0^{(0)}, \tilde{T}_0^{(1)}, \cdots, \tilde{T}_0^{(k-1)})$,
$(c_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)})$
$(c_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)})$

$\vdots$

$(c_0^{(k-1)}, y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_{n-1}^{(k-1)})$ を出力する.

2.3.4 署名検証

$(j = 0, 1, \cdots, k-1)$ について以下を実行する.

```
$c_0 := c_0^{(j)}, y_0 := y_0^{(j)}, \cdots, y_{n-1} := y_{n-1}^{(j)}$.

$(i = 0, 1, \cdots, n-1)$ について
以下を順次計算する.
    $T_i = E_i^j(T_i^{(0)})$
    $\tilde{T}_i = \tilde{E}^i(T_0^{(j)})$
    $x_i = g^{y_i}(T_i\tilde{T}_i)^{c_i} v_i \bmod p_i$
    $c_{i+1} = H_i(x_i \| T \| m)$
等式 $c_0 = c_n$ が成り立つか確認する.
```

全ての等式 $c_0 = c_n$ が成り立てば署名を受理し, そうで

なければ受理しない.

3 提案方式のプロトコル

本章では, [1] の "just"k-out-of-n 署名を提案方式に拡張する. "just"k-out-of-n 署名はちょうど k 人の署名者がいることを検証者が確認できる. 一方, 提案方式は n 人のグループのユーザの内で, l 人以上 m 人以下 $(1 \le l < m \le n)$ の署名者がいることを検証者が確認できる. これは "just"l-out-of-n 署名を作成する処理 (処理(1)とする) と, "just"1-out-of-n 署名を $m-l$ 回作成する処理 (処理(2)とする) を行うことによって, 署名者が最低でも l 人いることが確認でき, 加えて最大で $m-l$ 人の別の署名者がいることがわかる. すなわち処理 (1) によって署名者数の下限を作り, 処理 (1) と処理 (2) を合わせることによって, 署名者数の上限を作る. 以上の処理によって提案方式を実現し, 最低 l 人, 最高で m 人が署名作成に協力していることがわかる. また以降, 本方式を (l,m,n) 署名と呼ぶ.

3.1 準備

提案方式のパラメータは "just"k-out-of-n 署名のそれに従う. 以下の署名作成 step4 以降では "just "1-out-of-n 署名を作成するので, $k=1$ 固定である. よって $T_0^{(0)}$ の (0) は不要である. 従って step4 以降の $T_i^{<j>}$ というパラメータは "just"1-out-of-n 署名を $m-l$ 回作成するうちの, j 回目のユーザ i のためのパラメータ T を意味するものとする.

3.2 署名作成

ここでは "just"l-out-of-n 署名と, "just"1-out-of-n 署名を $m-l$ 回行う処理を合わせて, l 人以上 m 人以下 $(1 \le l < m \le n)$ における (l,m,n) 署名を作成する.
また署名作成に必要な前処理を step1, 2, 4 で行う.

step1 (共通鍵暗号パラメータ T の初期化)
2.2.2 章の署名作成 step1 と同様にして,
$(T_0^{(0)}, T_1^{(0)}, \cdots, T_{n-1}^{(0)})$ を得る.

step2 (共通鍵暗号パラメータ $\tilde{T}$ の初期化)
2.3.2 章の署名作成 step2 において k=l として,
$(\tilde{T}_0^{(0)}, \tilde{T}_0^{(1)}, \cdots, \tilde{T}_0^{(l-1)})$ を得る,

step3 ("just"l-out-of-n 署名を行う)
$T = T_0^{(0)} \| T_1^{(0)} \| \cdots \| T_{n-1}^{(0)} \| \tilde{T}_0^{(0)} \| \tilde{T}_0^{(1)} \| \cdots \| \tilde{T}_0^{(l-1)}$
とする.

```
(j = 0,1,⋯,l−1) について, 以下を繰り返す.
[ $i := f^{-1}(j)$    %ユーザの決定

  ユーザ $i$ は以下を実行する
  $x_i = g_i^{r_i} \bmod p_i$
  $c_{i+1} = H_i(x_i \| T \| m)$
  ただし $r_i \in Z_{q_i}$ とする

  $(t = i+1, \cdots, n-1, 0, 1, \cdots, i-1)$ について
  [
    ユーザ $i$ は以下を順次計算する
    $T_t = E_t^j(T_t^{(0)})$
    $\tilde{T}_t = \tilde{E}^t(\tilde{T}_0^{(j)})$
    $x_t = g_t^{y_t}(T_t\tilde{T}_t)^{c_t} v_t \bmod p_t$
    $c_{t+1} = H_t(x_t \| T \| m)$
    ただし $y_t \in Z_{q_i}$ とする
  ]
  署名作成ユーザ $i$ は
  $y_i = r_i + s_i - c_i(\alpha_i\bar{\alpha}_i)$ を計算する

  $c_0^{(j)} := c_0$,
  $y_0^{(j)} := y_0, y_1^{(j)} := y_1, y_{n-1}^{(j)} := y_{n-1}$
]
```

step3 の結果として,
$(c_0^{(0)}, y_0^{(0)}, y_1^{(0)}, \cdots, y_{n-1}^{(0)})$,
$(c_0^{(1)}, y_0^{(1)}, y_1^{(1)}, \cdots, y_{n-1}^{(1)})$,
$\vdots$
$(c_0^{(l-1)}, y_0^{(l-1)}, y_1^{(l-1)}, \cdots, y_{n-1}^{(l-1)})$
を出力する

step4(共通鍵暗号パラメータ $T, \tilde{T}$ の初期化)
2.2.2 章の署名作成 step2, 2.3.2 章の署名作成 step3 と同様にして,
$(T_0^{<0>}, T_1^{<0>}, \cdots, T_{n-1}^{<0>})$,
$(T_0^{<1>}, T_1^{<1>}, \cdots, T_{n-1}^{<1>})$,
$\vdots$
$(T_0^{<m-l>}, T_1^{<m-l>}, \cdots, T_{n-1}^{<m-l>})$,
$(\tilde{T}_0^{<0>}, \tilde{T}_0^{<1>}, \cdots, \tilde{T}_0^{<m-l>})$ を得る.

step5("just"1-out-of-n 署名を $m-l$ 回行う)

```
(j = 0,1,⋯,m−l) について, 以下を繰り返す.
[
  本章の step3 において $j = 0$ として,
  "just"1-out-of-n 署名を作成する.
]
```

表 1: 性能の比較

署名の種類	共通鍵暗号関数の個数	署名長
just k-out-of-n 署名	$n+k$	$\lvert T_i^{(0)} \rvert + \lvert \tilde{T}_0^{(j)} \rvert + \lvert c_0^{(j)} \rvert + \lvert y_i^{(j)} \rvert \ (0 \le i < n), (0 \le j < k)$
提案方式	$n+m$	$\lvert T_i^{(0)} \rvert + \lvert \tilde{T}_0^{(j)} \rvert + \lvert c_0^{(j)} \rvert + \lvert y_i^{(j)} \rvert \ (0 \le i < n), (0 \le j < l)$ $\lvert T_i^{<j>} \rvert + \lvert \tilde{T}_0^{<j>} \rvert + \lvert c_0^{<j>} \rvert + \lvert y_i^{<j>} \rvert \ (0 \le i < n), (0 \le j < m-l)$

step5 の結果として,

$$(c_0^{<0>}, y_0^{<0>}, y_1^{<0>}, \cdots, y_{n-1}^{<0>})$$
$$(c_0^{<1>}, y_0^{<1>}, y_1^{<1>}, \cdots, y_{n-1}^{<1>})$$
$$\vdots$$
$$(c_0^{<m-l>}, y_0^{<m-l>}, y_1^{<m-l>}, \cdots, y_{n-1}^{<m-l>})$$

を出力する.

step6(署名出力)

$(T_i^{(0)}, \tilde{T}_0^{(j)}, c_0^{(j)}, y_i^{(j)})(0 \le i < n)(0 \le j < l)$

$(T_i^{<j>}, \tilde{T}_0^{<j>}, c_0^{<j>}, y_i^{<j>})(0 \le i < n)(0 \le j < m-l)$

を出力する.

3.3 署名検証

$(j = 0, 1, \cdots, l-1)$ について以下を実行する.

```
    $c_0 := c_0^{(j)}, y_0 := y_0^{(j)}, \cdots, y_{n-1} := y_{n-1}^{(j)}.$

    $(i = 0, 1, \cdots, n-1)$ について
    以下を順次計算する.
        $T_i = E_i^j(T_i^{(0)})$
        $\tilde{T}_i = \tilde{E}^i(\tilde{T}_0^{(j)})$
        $x_i = g^{y_i}(T_i\tilde{T}_i)^{c_i} v_i \bmod p_i$
        $c_{i+1} = H_i(x_i \| T \| m)$
    等式 $c_0 = c_n$ が成り立つか確認する.
```

$(j = 0, 1, \cdots, m-l-1)$ について以下を実行する.

```
    $c_0 := c_0^{<j>}, y_0 := y_0^{<j>}, \cdots, y_{n-1} := y_{n-1}^{<j>}.$

    $(i = 0, 1, \cdots, n-1)$ について
    以下を順次計算する.
        $T_i = E_i^j(T_i^{(0)})$
        $\tilde{T}_i = \tilde{E}^i(\tilde{T}_0^{(j)})$
        $x_i = g^{y_i}(T_i\tilde{T}_i)^{c_i} v_i \bmod p_i$
        $c_{i+1} = H_i(x_i \| T \| m)$
    等式 $c_0 = c_n$ が成り立つか確認する.
```

全ての等式 $c_0 = c_n$ が成り立てば署名を受理し, そうでなければ受理しない.

4 提案方式の考察

本章では署名長, 安全性, 計算量について考察する. 安全性については, 理想的な共通鍵暗号関数の仮定において, 提案方式の基礎となる [1] 方式と同程度である.
共通鍵暗号の個数, 署名長についてはオリジナル方式である "just"k-out-of-n 署名との比較を表 1 に示す. 計算量については, step4, step5 において "just" 1-out-of-n 署名を作成する際に, 本稿では一般的な "just"k-out-of-n 署名の $k=1$ のケースとして, "just" 1-out-of-n 署名を作成した. しかし $k=1$ 専用のプロトコルを用意すれば, 署名作成時にパラメータ T の削減ができる. 従って計算量, 署名長の削減が可能であるが, 本稿では簡単化のため "just"k-out-of-n 署名作成のプロトコルをそのまま使用した.

5 まとめ

本稿では, 従来方式の少なくとも k 人や, ちょうど k 人による署名を, [1] 方式の "just"k-out-of-n 署名を利用して, l 人以上 m 人以下による, (l, m, n) 署名に拡張した.

参考文献

[1] 岡本健, 岡本栄司, "just" k-out-of-n 署名方式の提案, CSEC22-21, pages 151-156, 2003.

[2] R.Rivest, A.Shamir and Y.Tauman, How to leak a secret, In Asiacrypt '01, LNCS No.2248, pages 552-565. Springer-Verlag, 2001.

[3] R.Cramer, I.Bjerre Damgard and B.Schoenmakers, Proof of partial knowledge and simplified design of witness hiding protocols, In Crypto '94, LNCS No.839, pages 174-187, Springer-Verlag, 1994.

[4] 大久保美也子, 阿部正幸, 鈴木幸太郎, 辻井重男, 証明長が短い 1-out-of-n 証明, SCIS 2002, pages 189-193, 2002.

[5] E.Bresson, J.Stern and M.Szydlo, Threshold ring signatures and applications to ad-hoc groups, In Crypto '02, LNCS No.2442, pages 465-480, Springer-Verlag, 2002.

[6] 菊池浩明, 多田美奈子, 中西祥八郎, Ring signature に基づいた k-out-of-n 証明の提案, CSS 2002, pages 83-87, 2002.

[7] 菊池浩明, 多田美奈子, 中西祥八郎, リング署名プロトコルにおける署名者開示, CSEC20-27, pages 149-153, 2003.

[8] 須賀祐治, 岩村恵市, 否認機能を持つリング署名方式, SCIS 2003, pages 435-439, 2003.

SCIS 2004 The 2004 Symposium on
Cryptography and Information Security
Sendai, Japan, Jan.27-30, 2004
The Institute of Electronics,
Information and Communication Engineers

1-out-of-n 証明から k-out-of-n 証明への引き上げ方法
How to convert 1-out-of-n proof into k-out-of-n proof

椎名 信行* 岡本 健† 岡本 栄司†
Nobuyuki SHIINA Takeshi OKAMOTO Eiji OKAMOTO

あらまし 本稿では, 既存の 1-out-of-n 証明をそのまま用いて k-out-of-n 証明を実現する方法を提案する. その際, 1-out-of-n 証明の一つとして Ring 署名方式を採用しているが, 他の 1-out-of-n 証明においても提案方式は適用可能である. これまでの研究については, Ring 署名方式に基づく k-out-of-n 証明として Bresson ら [6] や菊池ら [4] が提案している方式があるが, いずれも既存の Ring 署名システムをそのまま用いることはできず, さらに, そのアルゴリズムは Ring 署名方式にしか適用出来ない. また, 提案方式は (k,n) 閾値秘密分散法にも応用可能であるため, その応用方法について述べ, 最後により効率の良い (k,n) 閾値秘密分散法を提案する.

キーワード 1-out-of-n 証明, k-out-of-n 証明, Ring 署名, 秘密分散法

1 はじめに

1-out-of-n 証明では n 人で構成されるグループのうち, 少なくとも 1 人のメンバが署名をするものであり, 検証者はグループ内の誰かが署名をしたことを確認できるが, 実際に署名をしたのが誰であるかまでは特定できない. それに対し, k-out-of-n 証明では n 人中, 少なくとも k 人が協力して署名を行なうものであり, 検証者は署名がそのグループ内の k 人によるものであることを確認できるが, k 人のうち誰一人たりとも特定することはできない. このような特徴を持つ両証明方式 [1] は内部告発者の保護に非常に有益であり, さらに, k-out-of-n 証明は議員や役員の罷免といったグループ内の何人が同意しているかを検証したいような状況に応用でき, 両証明方式とも実社会において今後益々求められる機能を提供する.

また, ある秘密情報 S を安全に保管するような場合を考えてみる. このとき考えられる問題点として, S の紛失や敵による S の盗難が挙げられる. 前者の問題点には S のコピーを作ることで対処できるが, 逆に後者の問題点に対する心配が大きくなるという相反する問題が存在する. この相反する問題を解決する手段として, S を n 個の秘密情報に分割して保管し, k 個の秘密情報が集まると元の S を復元できるという (k,n) 閾値秘密分散法がある. 代表的な例に Shamir により提案された多項式補間を利用した方式があるが, 提案方式を用いても実現可能であることを後述する.

これまでの研究において, 1-out-of-n 証明としては, グループ署名 [1] や Ring 署名と呼ばれる署名方式がある. 前者には特権を持つ管理者が存在し, 管理者をもとにメンバの登録, グループの公開鍵の作成といったセットアップ作業が必要であり, また, 管理者による署名者の特定が可能であるという特徴を持つ. それに対し, 後者には管理者が存在せず, セットアップ作業が必要なく, Ring を構成するメンバを署名者が任意に選べるが, メンバは第三者に自分が署名者でないことを証明することはできない [2]. Ring 署名にはいくつかの種類の方式が提案されており, Rivest らによる落とし戸付き一方向性関数を用いた方式 [2] や, 大久保らによる離散対数問題に基づく方式 [3] 等があるが, 本稿では [3] の方式を採用する. 一方, k-out-of-n 証明としては, 桑門らによる方式 [5] や, Ring 署名に基づく方式が Bresson ら [6] や菊池ら [4] により提案されている. しかしながら, 後者のいずれも Ring 署名方式にのみ有効であり, 他の 1-out-of-n 証明には適用できない. それに対し, 提案方式では他の 1-out-of-n 証明においても同様に k-out-of-n 証明が実現可能である. なぜなら, 従来方式ではいずれも Ring 署名方式のチャレンジをハッシュで連結していく箇所を変更して k-out-of-n 証明を実現していたが, 提案方式では 1-out-of-n 証明の

* 筑波大学 第三学群情報学類, 〒 305-8573 茨城県つくば市天王台 1-1-1. College of Information Sciences, Third Cluster of Colleges, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan.
† 筑波大学 電子・情報工学系, 〒 305-8573 茨城県つくば市天王台 1-1-1. Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan.

[1] [9] で署名者の人数が at least ではなく, just である方式が提案されている.

[2] 須賀らによって否認機能を持つ Ring 署名方式 [7] が提案されてはいるが課題が多い.

アルゴリズムはそのまま用い, メンバ構成のみを変更して k-out-of-n 証明を実現しているからである. また, 先程述べたように, 提案方式は (k,n) 閾値秘密分散法への応用が可能である.

本稿の構成としては, 第2節で [3] を, 第3節で [4] を説明する. 第4節では提案方式による k-out-of-n 証明を記し, 従来方式との比較・検討を行なう. 第5節では提案方式の秘密分散法への応用方法について述べ, 第6節でより効率の良い秘密分散法について記す. 最後に, 第7節でまとめる.

2 Ring 署名方式

本節では, 提案方式の基本となる大久保らによる離散対数問題に基づく Ring 署名方式 [3] について述べる.

2.1 準備

n 人で Ring が構成され, i 番目のメンバ U_i がメッセージ m に署名を行なうとする. 各メンバ $U_j (j=0,\cdots,n-1)$ において, p_j, q_j は $q_j \mid p_j - 1$ を満たす大きな素数とし, g_j を p_j の位数 q_j の部分群の生成元とする. $x_j \in \mathbb{Z}_{q_j}$ を秘密鍵, $y_j = g_j^{x_j} \bmod p_j$ を g_j, p_j, q_j と共に公開鍵とする. また, H_j をハッシュ関数 $H_j : \{0,1\}^* \to \mathbb{Z}_{q_{j+1}}$ とする. (ただし, 添字は $\bmod\, n$ とする)

2.2 署名生成

step1　署名者 U_i について,

$$\alpha \in_U \mathbb{Z}_{q_i}$$
$$T_i := g_i^{\alpha} \bmod p_i$$
$$c_{i+1} := H_i(m \| T_i)$$

を計算する.

step2　署名者以外 $U_j (j = i+1,\cdots,n-1,0,\cdots,i-1)$ について,

$$r_j \in_U \mathbb{Z}_{q_j}$$
$$T_j := g_j^{r_j} y_j^{c_j} \bmod p_j$$
$$c_{j+1} := H_j(m \| T_j)$$

を順次計算する.

step3　(秘密鍵 x_i を知る) 署名者 U_i について,

$$r_i := \alpha - x_i c_i \bmod q_i$$

を計算する.

step4　$(c_0, r_0, \cdots, r_{n-1})$ を m に対する署名として出力する.

Ring 署名方式は Schnorr 署名方式の逐次的な結合と見なせる. まず, step1 で署名者 U_i がコミットメント α を用意し, ハッシュによりチャレンジを次段へ与える. step2 では前段からのチャレンジを受け取り, 同様に次段へとハッシュでチャレンジを連結していく. 最後に, step3 で署名者 U_i が秘密鍵 x_i によりチャレンジ c_i に対するレスポンスを求める.

また,

$$\begin{aligned} T_i \equiv g_i^{\alpha} &\equiv g_i^{r_i} y_i^{c_i} \\ &\equiv g_i^{\alpha - x_i c_i} (g_i^{x_i})^{c_i} \\ &\equiv g_i^{\alpha} \pmod{p_i} \end{aligned}$$

より, x_i を知る署名者 U_i は Ring を閉じることが可能であることがわかる.

2.3 署名検証

step1　各メンバ $U_j (j = 0,\cdots,n-1)$ について,

$$T_j := g_j^{r_j} y_j^{c_j} \bmod p_j$$
$$c_{j+1} := H_j(m \| T_j)$$

を順次計算する.

step2　$c_0 = c_n$ なら受理し, そうでなければ棄却する.

$c_0 = c_n$ ならば, 少なくとも 1 人のメンバが秘密鍵を用いて Ring を閉じたことがわかる.

3 従来方式による k-out-of-n 証明

本節では, 菊池らによる k-out-of-n 証明 [4] について述べる.

3.1 準備

本節で使用する記号を以下の表 1 に示す. その他の記号は 2.1 と同様である. (ただし, 添字は $\bmod N$ とする)

表 1 : 本節における記号の定義

記号	説明
k	署名者の人数
R	Ring の構成メンバの集合 (要素数 n)
S	署名者の集合 ($S \in R$) (要素数 k)
R'	新たに作成される Ring
N	R' の要素数

3.2 署名生成

step1　R の大きさ k の部分集合 $A = \{U_{i_0}, \cdots, U_{i_{k-1}}\}$ の全てに対して必ず

$$U_{i_0} = U_{j_l}, \cdots, U_{i_{k-1}} = U_{j_{l+k-1}}$$

が成立する Ring$R' = \{U_{j_0}, \cdots, U_{j_N - 1}\}$ を作成し, 公開する.

step2　$l = t, \cdots, t+k-1$ $(S = \{U_{j_t}, \cdots, U_{j_{t+k-1}}\})$ について,

$$\alpha_l \in_U \mathbb{Z}_{q_{j_l}}$$
$$T_l := g_{j_l}^{\alpha_l} \bmod p_{j_l}$$

を計算する.

step3　以下の処理を行なう.

$$c_{t+k} := H_{j_{t+k-1}}(m \| T_t \| \cdots \| T_{t+k-1})$$

step4　$l = t+k, \cdots, N-1, 0, \cdots, t+k-2$ について,

$$r_l \in_U \mathbb{Z}_{q_{j_l}}$$
$$T_l := g_{j_l}^{r_l} y_{j_l}^{c_l} \bmod p_{j_l}$$
$$c_{l+1} := H_{j_l}(m \| T_{l-k+1} \| \cdots \| T_l)$$

を順次計算する.

step5　$l = t, \cdots, t+k-1$ について,

$$r_l := \alpha_l - x_{j_l} c_l \bmod q_{j_l}$$

を計算する.

step6　$(c_0, \cdots, c_{k-1}, r_0, \cdots, r_{N-1})$ を署名として出力する.

この方式は, ハッシュの中身を多重にし, k 箇所で Ring を閉じることにより k-out-of-n 証明を実現している. そのためには, Ring 署名の特性上, 署名者が隣合った位置に存在する Ring でなくてはならない. したがって, step1 でそのような Ring を用意し, 公開する必要がある. step2 では, k 個のコミットメントを用意し, step3 でハッシュによりチャレンジを次段へ渡す. step4 では, 前段からチャレンジを受け取り, 同様に次段へとハッシュでチャレンジを連結していく. 最後に, step5 で署名者が各々秘密鍵を用いてレスポンスを求める.

3.3 署名検証

step1　$l = 0, \cdots, k-1$ について,

$$T_l := g_{j_l}^{r_l} y_{j_l}^{c_l} \bmod p_{j_l}$$

を計算する.

step2　以下の処理を行なう.

$$c'_k := H_{j_{k-1}}(m \| T_0 \| \cdots \| T_{k-1})$$

step3　$l = k, \cdots, N-1, 0, \cdots, k-2$ について、

$$T_l := g_{j_l}^{r_l} y_{j_l}^{c_l'} \bmod p_{j_l}$$
$$c'_{l+1} := H_{j_l}(m \| T_{l-k+1} \| \cdots \| T_l)$$

を順次計算する.

step4　$c_0 = c'_0, \cdots, c_{k-1} = c'_{k-1}$ ならば受理し, そうでなければ棄却する.

$c_0 = c'_0, \cdots, c_{k-1} = c'_{k-1}$ ならば, 少なくとも k 人のメンバが秘密鍵を用いて k 箇所で Ring を閉じたことがわかる.

4 提案方式による k-out-of-n 証明

本節では, 第2節で紹介した大久保らによる Ring 署名方式 [3] を k-out-of-n 証明に引き上げる方法について説明する. 提案方式の特徴は, 従来方式とは違い, [3] の方式のアルゴリズムには一切変更を加えず, Ring のメンバ構成のみを変えて署名を行なうところにある. そのため, 既存の Ring 署名システムをそのまま用いて k-out-of-n 証明が実現可能となる. また, ベースとなる 1-out-of-n 署名に一切変更を加えないため, 他の 1-out-of-n 証明にも同様に応用でき, flexible な署名体系と言える. なお, 本節で扱う記号は 2.1, 3.1 と同様とする.

4.1 署名生成

step1　R の大きさ $n-k+1$ の部分集合 $B = \{U_{i_0}, \cdots, U_{i_{n-k}}\}$ を全て作り, 順に $R_0, \cdots, R_{{}_nC_{k-1}-1}$ とする.

step2　各 Ring $(R_0, \cdots, R_{{}_nC_{k-1}-1})$ に対して, [3] による署名を行なう.

step3　各 Ring から出力された署名全体を署名として出力する.

$n-k+1 = n-(k-1)$ より, step1 では R から全ての $k-1$ 人の組み合わせを除いた Ring を作成している. このとき作成された全ての Ring $(R_0, \cdots, R_{{}_nC_{k-1}-1})$ について考えてみると, 署名者が k 人未満であるならばその除かれる $k-1$ 人の組み合わせによって全ての署名者が除かれてしまう Ring が少なくとも 1 つは必ず存在するはずである. そのような署名者のいない Ring では, 落とし戸情報を持つメンバがいないため, Ring を閉じることが出来ない. したがって, step2 で少なくとも一つの Ring に対して署名に失敗し, step3 で正当な署名を作成することが不可能となる.

4.2 署名検証

step1　各 Ring に対して, 対応する署名の [3] による検証を行なう.

step2　全ての Ring が [3] の署名検証をパスしたならば受理し, そうでなければ棄却する.

表 2 : 各メンバが持つ秘密情報 (I)

U_0	S_0	S_1	S_2	S_3	S_4	S_5				
U_1	S_0	S_1	S_2				S_6	S_7	S_8	
U_2	S_0			S_3	S_4		S_6	S_7		S_9
U_3		S_1		S_3		S_5	S_6		S_8	S_9
U_4			S_2		S_4	S_5		S_7	S_8	S_9

全ての Ring が署名検証をパスしたならば, 全ての Ring に少なくとも 1 人の署名者が含まれていることになり, 署名者が k 人以上であることが 4.1 より証明される. 注意点としては, step1 で各 Ring に対して, 対応する署名の検証を行なうわけだが, そのためには署名と Ring との対応関係を知る手段が必要となる. 対応関係を署名に添付することも可能だが, 署名長が長くなり, 望ましくない. 別の手段としては, 署名生成時の $R_0, \cdots, R_{{}_nC_{k-1}-1}$ の作成手順を決めておき, 検証者がその手順から各リングと署名の対応関係を求める方法が考えられる. もしくは, [4] のように $R_0, \cdots, R_{{}_nC_{k-1}-1}$ を公開する方法が挙げられる.

4.3 考察

4.3.1 安全性

提案方式は, 既存の 1-out-of-n 証明に一切変更を加えずそのまま用いることから, 安全性はベースとなる 1-out-of-n 証明の安全性に依存する. すなわち, 上記の [3] による k-out-of-n 証明では, 離散対数問題の困難性とハッシュ関数の一方向性に依存するため安全であると言える.

また, 提案方式の匿名性について考察する. 提案方式では, 任意に選んだメンバが署名者である確率は k/n で, S が特定される確率は $1/{}_nC_k$ となり, どれも一様である. さらに, k 人の署名者のうち, 1 人が判明したとしても, 残りの $k-1$ 人の特定に関する情報は得られない. したがって, S の特定は困難である.

4.3.2 効率

提案方式の署名長は, ${}_nC_{k-1}(|q_j| + \sum_{j=0}^{n-1} |q_j|)$ bit となる. 一方, [6] の方式では, 約 $2^k \lceil \log_2 n \rceil (\sum_{j=0}^{k-1} l_j + \sum_{j=0}^{n-1} l_j)$ bit となる. なお, [6] の方式は, 落とし戸付き一方向性関数に基づくものであり, l_j はその入力 bit 数を表す. また, [4] の方式の署名長は, 3.2 の step1 のアルゴリズムが明記されていないため, 具体的な署名長は不明である.

より具体的な値として, $|q_j| = 160$, $l_j = 1024$ ($j = 0, \cdots, n-1$) としたとき [3], 提案方式の署名長は $160 \times {}_nC_{k-1}(1+n)$ bit となり, [6] の方式での署名長は約 $1024 \times 2^k \lceil \log_2 n \rceil (k+n)$ bit となる. ここで, 提案方式と [6] の署名長を比較してみると, k が n に近いとき, 明らかに提案方式の方が署名長が短くなり, 効率が良い. 逆に, k が $n/2$ に近い時, 提案方式は最も署名長が長くなり, 効率は悪い.

5 提案方式による秘密分散法

本節では, 提案方式の (k,n) 閾値秘密分散法への応用方法について説明する.

5.1 アルゴリズム

step1 秘密情報 S を以下のように分割する.

$$S = S_0 \oplus \cdots \oplus S_{{}_nC_{k-1}-1}$$

step2 大きさ $n-k+1$ の部分集合 $C = \{U_{i_0}, \cdots, U_{i_{n-k}}\}$ を全て作り, 順に $G_0, \cdots, G_{{}_nC_{k-1}-1}$ とする. ただし, U_i はメンバを表す.

step3 $j = 0, \cdots, {}_nC_{k-1}-1$ について, G_j に S_j を配布する.

例えば, $n = 5, k = 3$ のとき, ${}_5C_{3-1} = 10$ より, 秘密情報 S を $S = S_0 \oplus \cdots \oplus S_9$ のように分割する. 次に, 大きさ $3 (= 5-3+1)$ のグループを全て作成し, 分割した情報 $S_0, \cdots, S_9$ を以下のように順に各グループに配布する. その結果, 各メンバは表 2 のような情報を保持することになる.

$S_0 \rightarrow (U_0, U_1, U_2)$	$S_1 \rightarrow (U_0, U_1, U_3)$
$S_2 \rightarrow (U_0, U_1, U_4)$	$S_3 \rightarrow (U_0, U_2, U_3)$
$S_4 \rightarrow (U_0, U_2, U_4)$	$S_5 \rightarrow (U_0, U_3, U_4)$
$S_6 \rightarrow (U_1, U_2, U_3)$	$S_7 \rightarrow (U_1, U_2, U_4)$
$S_8 \rightarrow (U_1, U_3, U_4)$	$S_9 \rightarrow (U_2, U_3, U_4)$

表 2 からわかるように, どの 3 人未満のメンバが協力したとしても少なくとも 1 つの情報が足りないため, 元の秘密情報 S を復元することは出来ない. しかしながら, 3 人以上の協力があるとすべての情報を得ることができ, S を復元できる. これは 4.1 より明らかである. また, 3 人未満の協力では S に関する一切の情報が漏れないため, 完全秘密分散であると言える.

[3] 落とし戸付き一方向性関数に RSA を用いるとする.

表 3：各メンバが持つ秘密情報 (II)

U_0	S'_0			S'_3			S'_6		
U_1		S'_1		S'_3			S'_6		
U_2			S'_2		S'_4		S'_6		
U_3			S'_2			S'_5		S'_7	
U_4			S'_2			S'_5			S'_8

5.2　考察

5.2.1　安全性

4.1 から k 人未満の協力では $S_0, \cdots, S_{{}_nC_{k-1}-1}$ のうち少なくとも 1 つの情報が手に入らず，秘密情報 S を復元できないことがわかる．また，$S = S_0 \oplus \cdots \oplus S_{{}_nC_{k-1}-1}$ より，$S_0, \cdots, S_{{}_nC_{k-1}-1}$ の全ての情報が手に入らない限り S に関する情報は一切漏洩しない．

5.2.2　効率

各メンバが持つ情報量は $({}_nC_k - {}_{n-1}C_k)|S|$ bit となり，全体の情報量は $n({}_nC_k - {}_{n-1}C_k)|S|$ bit となる．これは，任意の $k-1$ 人が協力した時，秘密情報 S を復元するのに必要な情報は $S_0, \cdots, S_{{}_nC_{k-1}-1}$ のうちどれか 1 つだけにもかかわらず，その 1 つの情報を得るのに $({}_nC_k - {}_{n-1}C_k)$ 個の情報を持つ k 人目の協力が必要となり，無駄な情報が多く生じていることになる．

5.2.3　利点

あるメンバが持つ情報から適当な情報を取り除くことにより，ある k 人の組み合わせでは秘密情報 S を復元できないようにすることが可能となる．つまり，メンバ毎に重みを付け，メンバ間の差別化が可能である．

6　効率の良い秘密分散法

本節では，5.2.2 で指摘した問題点を改善したより効率の良い秘密分散法について述べる．

6.1　基本方針

5.1 では 1 つのブロック内で秘密情報 S を分割し，各メンバにその分割した情報を順に配布することにより秘密分散を行なっていたが，これを複数のブロックに分け，各ブロック毎に対して秘密分散を行なう時を考える．例えば，5 節の例と同じく (3,5) 閾値秘密分散を行なうとすると，まず以下のように S を 3 つの大きさ 3 のブロックに分割し，表 3 のようにその分割した情報を各メンバに配布する方法が考えられる．

$$\begin{aligned} S &= S'_0 \oplus S'_1 \oplus S'_2 \\ &= S'_3 \oplus S'_4 \oplus S'_5 \\ &= S'_6 \oplus S'_7 \oplus S'_8 \end{aligned}$$

このとき，表 3 からわかるように，どの 3 人未満が協力したとしても (S'_0, S'_1, S'_2), (S'_3, S'_4, S'_5), (S'_6, S'_7, S'_8) のどの組み合わせも手に入らず，S を復元できない．しかしながら，少なくとも 3 人が協力することにより，(S'_0, S'_1, S'_2), (S'_3, S'_4, S'_5), (S'_6, S'_7, S'_8) のいずれかの組み合わせがわかり，S を復元できる．

6.2　考察

表 2 と表 3 を比較してみると，1 人当たりが持つ情報量は，前者が $6|S|$ bit に対し，後者が $3|S|$ bit となり，情報量が半減したことになる．今回の例では，1 つ当たりのブロックの大きさを 3 としているが，それ以上の大きさのブロックに分割した場合にブロック数が少なくて済み，効率が良くなる可能性も捨て切れない [4]．さらに，各々のブロックの大きさが同じである必要もなく，各ブロックの大きさが異なる場合も考慮する必要がある．また，(k,n) の値によっては 5 節の方式の方が情報量が少なくなる可能性も考えられる．これらの点についての研究は今後の課題とする．

7　おわりに

本稿では，1-out-of-n 証明を k-out-of-n 証明に引き上げる方法について示した．その際，1-out-of-n 証明の例として Ring 署名方式を採用したが，他の 1-out-of-n 証明においても同様に k-out-of-n 証明に引き上げることが可能である．本稿ではさらに，その提案方式の (k,n) 閾値秘密分散法への応用方法とその課題点を示し，より効率の良い秘密分散が可能である例を示した．

参考文献

[1] D.Chaum and E.van Heyst, "Group Signatures ", EUROCRYPT 1991, pp.257-265, 1991.

[2] Ronald L.Rivest, Adi Shamir and Yael Tauman, "How to Leak a Secret ", ASIACRYPT 2001, LNCS 2248, pp.552-565, 2001.

[3] 大久保 美也子，阿部 正幸，鈴木 幸太郎，辻井 重男，「証明長が短い 1-out-of-n 証明」，SCIS 2002, pp.189-193, 2002.

[4] 大きさが k より小さいブロックには出来ない．

[4] 菊池 浩明, 多田 美奈子, 中西 祥八郎, 「Ring Signature に基づいた k-out-of-n 証明の提案」, CSS 2002, pp.83-87, 2002.

[5] 桑門 秀典, 田中 初一, 「署名者の匿名性を有するディジタル署名方式」, CSEC18-35, pp.239-244, 2002.

[6] Emmanuel Bresson, Jacques Stern and Michael Szydlo, "Threshold Ring Signatures and Applications to Ad-hoc Groups ", Crypto 2002, pp.465-480, 2002.

[7] 須賀 祐治, 岩村 恵市, 「否認機能を持つリング署名方式」, SCIS 2003, pp.435-439, 2003.

[8] 樋口 裕樹, 菊池 浩明, 中西 祥八郎, 「Forward-Secure な Ring 署名スキーム」, SCIS 2003, pp.423-427.

[9] 岡本 健, 岡本 栄司, 「"just "k-out-of-n 署名方式の提案」, CSEC22-21, pp.151-156, 2003.

SCIS 2004 The 2004 Symposium on Cryptography and Information Security
Sendai, Japan, Jan.27-30, 2004
The Institute of Electronics, Information and Communication Engineers

An IBE Scheme to Exchange Authenticated Secret Keys

Routo Terada [*] Waldyr Dias Benits Jr. [†] Eiji Okamoto [‡]

Abstract— We present a variant of the Boneh & Franklin Identiy-based Encryption IBE scheme to derive an authenticated symmetric key-exchange protocol, when combined with a signature scheme. Our protocol uses IBE as a secure channel to establish a symmetric key between two users and, after that, further communication can be done by symmetric cryptography, much faster than pairing-based cryptography.

Keywords: Identity-based cryptosystem, pairing, bilinear map, elliptic curve cryptography, key-exchange protocol.

1 Introduction

The concept of identity-based cryptography was first proposed in 1984 by Adi Shamir [10]. In his paper, Shamir presented a new model of asymmetric cryptography in which the public key of any user is a characteristic that uniquely identifies himself/herself, like an e-mail address. Since the public keys are not random numbers, digital certificates are needless. After Shamir's model, many researchers tried to propose a cryptographic scheme in this model, but only in 2001 an efficient one was proposed by Dan Boneh and Matthew Franklin [3], based on pairings.

However, the IBE scheme of Boneh & Franklin (which we call "BF's scheme" in the remainder of this paper) depends on a random number chosen by the sender of the message, and if this random number is not carefully chosen, the security of the scheme is seriously compromised. This occurs because the BF's scheme encrypts a message using an XOR function, and as proved in [4], XOR-based functions can be easily broken by a chosen-plaintext attack if the same key is used more than once.

In this paper, we propose a variant of the BF's scheme, replacing the XOR function by a symmetric encryption algorithm. With this change, BF's scheme can be used as a key-exchange protocol provided that it is combined with a signature scheme to guarantee mutual authentication of the parties involved.

There are similar key-exchange protocols based on pairings, among them we can cite [5, 6, 9, 11, 12, 8]. But they are not based on "BF's scheme". [8] is a key-sharing scheme, not a key-exchange one.

This paper is organized as follows: Section 2 summarizes ID-based cryptosystems. Section 3 reviews the IBE BF's scheme and shows the need of a carefully chosen random number. Section 4 presents our variant of the BF's scheme, for the key-exchange purpose. Section 5 reviews a signature scheme to be used together with a modified Boneh & Franklin IBE, so as to guarantee authentication of the two parties in the protocol (a.k.a. mutual authentication). Finally, section 6 concludes the paper and proposes further research.

2 Identity-based cryptosystems

2.1 Bilinear maps

Let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order, in which the discrete logarithm problem (DLP) is assumed to be hard. Concretely, we can think of G_1 as a group of points on an elliptic curve over F_q, and G_2 as a subgroup of the multiplicative group of a finite field $F^*_{q^k}$ for some $k \in Z^*$. Let P be a generator of G_1 and let $e : G_1 \times G_1 \longrightarrow G_2$ be a mapping with the following properties:

1. bilinearity: A mapping e is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and for all $a, b \in F_q$, where F_q is a finite field of order q;
2. non-degeneracy: A mapping is non-degenerate if exists $Q \in G_1$ so that $e(P, Q) \neq 1$, that is, the mapping does not map all pairs in $G_1 \times G_1$ to the identity in G_2;
3. Computability: A mapping is efficiently computable if $e(P, Q)$ can be computed in polynomial-time for all $P, Q \in G_1$.

Examples of pairings that satisfy these properties are the Weil pairing and the Tate pairing. Due to many improvements on its computation [1, 2], we consider the Tate pairing more efficient than Weil pairing.

2.2 Key generation

In ID-based cryptosystems we need a private key generator (PKG) which generates a pair of keys based on the identity of the user. After generating the keys, the PKG uses a secure channel to send the private key to the owner of the identity.

[*] University of S Paulo, Brazil - rt@ime.usp.br
[†] University of S Paulo, Brazil - waldyrbenits@ip2.com.br
[‡] University of Tsukuba, Japan - okamoto@is.tsukuba.ac.jp

The process of generating keys is the following: let ID be an identity of a user Alice (e.g., her e-mail address). First, PKG computes Alice's public key (Q_{alice}), by mapping her identity to a point of the elliptic curve (using a hash function H_1) and then, PKG uses his master key $s \in F_q$ to compute Alice's private key (S_{alice}), multiplying Q_{alice} by s. This process is summarized below:

$$\begin{aligned} Q_{alice} &= H_1(ID) \quad (1) \\ S_{alice} &= sQ_{alice} \quad (2) \end{aligned}$$

Here, to strengthen the security model, we can view the hash function H_1 as a random oracle, defined as follows:

$$H_1 : \{0,1\}^* \longrightarrow G_1$$

3 The IBE BF's scheme

Let (Q_{bob}, S_{bob}) be a pair of Bob's identity-based keys; R_{PKG} be a standard public key of the entity that generates Bob's keys, so that $R_{PKG} = sP$, where s is PKG's master key and P a generator of G_1. Let m be a message that Alice intends to secretly send to Bob.

The original IBE scheme proposed by Dan Boneh and Matthew Franklin [3] is as follows:

3.1 Encryption

Alice chooses a random number $r \in F_q$ and computes ($\oplus$ represents exclusive-OR (XOR).):

$$\begin{cases} U &= rP \\ V &= m \oplus H_3(e(R_{PKG}, rQ_{bob})) \end{cases} \qquad (3)$$

and sends the ciphertext (U, V) to Bob.
To increase security, we can view the hash function H_3 as a random oracle, defined as follows:

$$H_3 : G_2 \longrightarrow \{0,1\}^*.$$

Notice that Alice uses Bob's ID-based public key to encrypt m. To do that, she needs only to know his identity (e.g. Bob's e-mail address).

3.2 Decryption

Bob, after receiving (U, V) from Alice, performs the following computation to recover cleartext m:

$$m = V \oplus H_3(e(U, S_{bob})) \qquad (4)$$

Clearly, we can see that Bob uses his ID-based private key to find m.

3.3 Verification

Let us show how Bob is able to recover m:

$$\begin{aligned}
&V \oplus H_3(e(U, S_{bob})) \\
&= V \oplus H_3(e(rP, S_{bob})), && \text{as } U = rP \text{ by(3)} \\
&= V \oplus H_3(e(rP, sQ_{bob})), && \text{as } S_{bob} = sQ_{bob} \text{ by(2)} \\
&= V \oplus H_3(e(P, Q_{bob}))^{rs}, && \text{by bilinearity} \\
&= V \oplus H_3(e(sP, rQ_{bob})), && \text{by bilinearity} \\
&= V \oplus H_3(e(R_{PKG}, rQ_{bob})), && \text{as } R_{PKG} = sP \\
&= m, && \text{due to equation (3)} \\
&V = m \oplus H_3(e(R_{PKG}, rQ_{bob}))
\end{aligned}$$

3.4 Security of BF's scheme

Schemes based on pairings, like BF's scheme, depend not only on the hardness of DLP, but also on the hardness of a problem known as bilinear Diffie-Hellman problem (BDHP). First, we define BDHP as follows:

- BDHP – Given $(P, aP, bP, cP) \subset G_1$ compute $e(P, P).^{abc}$

We are going to show that, even if the DLP was hard, an adversary could obtain advantages if the BDHP was easy.

Suppose that Alice sent a message m to Bob, using BF's scheme. As we saw, Bob received (U, V) from Alice. If an adversary Charles intends to decrypt m, he must compute m using equation (4). However, as S_{bob} is unknown to Charles and assuming that DLP is hard, Charles fails to compute $e(S_{bob}, U)$ and cannot recover m.

Now, let us assume that Charles is able to easily solve the BDHP in the group chosen by Alice. We can see that Charles knows the following values:

- $R_{PKG} = sP$;
- $Q_{bob} = H_1(ID_{bob}) = hP$ (for some $h \in F_q$, because Q_{bob} is a point in the elliptic curve);
- $U = rP$, because we assumed that Charles had intercepted (U, V).

As DLP is hard, the values s, h and r are unknown to Charles, but assuming that he can solve the BDHP, if he knows sP, hP and rP he can compute $e(P, P)^{shr}$. Nevertheless:

$$\begin{aligned} e(S_{bob}, U) &= e(sQ_{bob}, rP) \\ &= e(shP, rP) \\ &= e(P, P)^{shr} \end{aligned}$$

Thus, assuming that the BDHP is easy, an adversary can compute correctly the pairing $e(S_{bob}, U)$ and recover m, even if the DLP is hard, that is, an adversary needs neither S_{bob} nor s to decrypt m.

We can guarantee the security in pairing-based cryptosystems by choosing appropriately the parameters k and q so that both the DLP and the BDHP are hard.

3.5 On the selection of a random number r

We saw in equation (3) that Alice must choose a random number r. In the original paper, the authors do not mention the importance of this choice. To increase the security of the scheme, Alice must choose a uniformly distributed and independent number, so as not to give any chance for an adversary to obtain advantage. For example, let us assume that Alice chooses r from a regular sequence r, $r+d$, $r+2d, \ldots$ (for some integer d). The adversary only needs to find one r and so, the others are easily predicted, and the security of the system is compromised.

An extreme situation occurs when Alice chooses the same r for two different messages. If she does that, any adversary can perform an attack similar to the one described in [4] — and summarized below — and recovers m.

Let us suppose that Alice sent messages m_1 and m_2 to Bob and she carelessly chose the same r. In the equation (3), we see that the value U will be the same, since it depends on r, but the value V will be different, as it depends on m. So, let V_1 and V_2 be the values computed by Alice when she sent m_1 and m_2, respectively.

Besides, as the values of $H_3(e(R_{PKG}, rQ_{bob}))$ for both messages (m_1 and m_2) remain constant since they do not depend on m, we can call them x.

Thus, let us rewrite the second equation of (3) as follows:

$$\begin{cases} V_1 &= m_1 \oplus x \\ V_2 &= m_2 \oplus x \end{cases}$$

If an adversary Charles intercepts values V_1 and V_2, he can compute:

$$\begin{aligned} V &= V_1 \oplus V_2 \\ &= (m_1 \oplus x) \oplus (m_2 \oplus x) \\ &= m_1 \oplus m_2 \end{aligned}$$

By knowing the value V, if an adversary obtains m_1 (without loss of generality), he can perform a known plaintext attack and gets m_2 successfully.

This weakness is because m is XOR-ed with a computed value that depends on a random number. To avoid this problem we propose next to replace the XOR function by a nonlinear function.

4 A key-exchange protocol

We propose a variant of the BF's scheme, so that it can be used as a key-exchange protocol. It is well known the existing symmetric cryptography is much faster than asymmetric one and, in practice, asymmetric cryptography is often used as a key exchange protocol to exchange a secret key which is then used to ensure secrecy between the parties.

We can do the same with BF's scheme by replacing the XOR function by a known symmetric encryption algorithm (e,g., 3-DES or AES). With this substitution, if Alice wants to communicate with Bob, she uses IBE only to exchange a secret key with Bob and then they can exchange secure messages using symmetric cryptography, which is known to be much faster than pairing-based cryptosystems.

The proposed variant is as follows: notice, by equations (3) and (4) that Alice and Bob compute the same value (Alice computes $H_3(e(R_{PKG}, rQ_{bob}))$ and we have proved that this value is equal to the one computed by Bob, $H_3(e(U, S_{bob}))$. Let us call this value k.

In our protocol, the value k, instead of being XOR-ed with a message by Alice to compute the encryption of m, (that is, V), it will be used as a symmetric encryption key between Alice and Bob. In this case, Alice does not have to carefully choose a uniformly distributed and independent random number for each message she intends to transmit as occurs in the original scheme. She only needs to choose one random number to exchange a key with Bob and after that, exchange as many encrypted messages as needed. If she later wants to change the secret key, she chooses another random number and establishes another key.

Moreover, as the key value depends on the hash function H_3, we can impose the hash value to be as large as we want. For example, we can define the hash function to map the pairing value to a 128-bit key, or 256-bit key or even a larger key.

The first step of our protocol is presented below:

4.1 Key establishment phase I

As in the encryption phase of BF's scheme, Alice chooses a random number $r \in F_q$ and computes:

$$\begin{cases} U &= rP \\ k &= H_3(e_t(rQ_{bob}, R_{PKG})) \end{cases} \tag{5}$$

and sends (U) to Bob.

4.2 Key establishment phase II

As in the decryption phase of BF's scheme, Bob will, after receiving (U) from Alice, perform the following computation to recover k:

$$k = H_3(e_t(S_{bob}, U)) \tag{6}$$

With this step, Alice is sure that only Bob can recover k, since only he knows the appropriate S_{bob}. However, for this protocol to be considered secure, Bob needs to be sure that the message was sent by Alice. Thus, to complete our protocol, it is necessary that Alice signs some information she sends to Bob, so as to prove that she is, in fact, Alice.

In the next section, we will see the complete key exchange protocol, combining BF's scheme with a signature scheme, to obtain mutual authentication.

5 Using a signature scheme together with BF's scheme

We saw in our key exchange protocol that Alice must sign some information she sends to Bob, in order to prove that she is Alice. In this section we will see an example of a signature scheme proposed by Florian Hess [7], that can be used together with BF's scheme.

5.1 Signature

If Alice wants to sign a message m, first she chooses a random number $t \in F_q$ and a random point $P_1 \in G_1^*$ and computes:

$$r = e(P_1, P)^t \tag{7}$$

Afterwards, she computes:

$$h = H_2(m||r) \tag{8}$$

and at last,

$$W = hS_{alice} + tP_1 \tag{9}$$

Now Alice's signature on m is (W, h). We can think of H_2 as a random oracle defined as follows:

$$H_2 : \{0,1\}^* \longrightarrow F_q$$

Notice by equation (9) that Alice uses her ID-based private key S_{alice} to compute the signature on m. See also that the sum in equation (9) represents a sum of points on an elliptic curve, since hS_{alice} and tP_1 are both points in G_1.

5.2 Verification

If Bob wants to verify that the signature comes from Alice, he must compute:

$$r = e(W, P) \cdot e(Q_{alice}, -R_{PKG})^h \tag{10}$$

After computing r, Bob accepts Alice's signature as valid if, and only if:

$$h = H_2(m||r) \tag{11}$$

Notice that only public parameters are used to verify the signature, meaning that anyone is able to perform such verification.

5.3 Proof

Now, we are going to prove that the equation (10) holds for a valid signature.

$$\begin{aligned}
& e(W, P) \cdot e(Q_{alice}, -R_{PKG})^h \\
&= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -R_{PK})^h \\
&= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -sP)^h \\
&= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -P)^{sh} \\
&= e(hS_{alice} + tP_1, P) \cdot e(sQ_{alice}, P)^{-h} \\
&= e(hS_{alice} + tP_1, P) \cdot e(S_{alice}, P)^{-h} \\
&= e(hS_{alice} + tP_1, P) \cdot e(-hS_{alice}, P) \\
&= e(hS_{alice} - hS_{alice} + tP_1, P) \\
&= e(tP_1, P) \\
&= e(P_1, P)^t \\
&= r \qquad \text{(due to equation (7))}
\end{aligned}$$

In Hess' scheme, we can see by equation (11) that a verifier needs to know the message m so as to perform the verification. There are other kinds of signature schemes, which include message recovery, in that the verifier by using sender's public key, is able to recover the message. Examples of the latter are RSA signatures.

If we use Hess' scheme to authenticate the sender Alice, our complete key exchange protocol will be as follows: after computing k by equation (5), Alice signs k with her private key S_{alice}, using k in Hess' scheme (equation 8) instead of m. After that, Alice sends U and (W, h) to Bob, where (W, h) is Alice's signature on k.

Bob, to receive the secret key k, first uses his private key S_{bob} to compute k (this way, Alice is sure that only the authentic Bob could decrypt correctly) and then he checks if (W, h) is a valid signature of Alice on k. If this verification is correct, they can start exchanging messages using k as a secret key; otherwise, he rejects k.

Alice can choose any signature scheme to combine with BF's scheme. If she decides to use a message-recovery signature scheme instead of Hess' scheme, she has to sign the value U, instead of k, otherwise anyone could recover the secret key k by only using Alice's public key. Bob, after receiving the signed value U, performs a signature verification to check if U comes from Alice. If not, he rejects U.

In our key exchange protocol, once Alice and Bob are authenticated and a key is exchanged, any message between them can be encrypted by an agreed upon symmetric algorithm (for example, Alice sends $C = C_{||}(\Updownarrow)$ and Bob computes $m = D_k(C)$, where D_k is the inverse of C_k, implemented by choosing secure algorithms such as AES, 3-DES *etc.*

6 Conclusions and further research

We have shown how to use IBE BF's scheme as a key exchange protocol by replacing the XOR operation by a symmetric algorithm and using it together with a signature scheme, so as to establish mutual authentication. This way, we avoid the dependency on a random number choice that can be exploited in the original protocol, if the sender does not carefully choose this random number. Moreover, our protocol allows the pairing-based scheme to be used only as a secure channel to transmit secret keys, and, after that, messages can be exchanged by symmetric cryptography, much faster than asymmetric cryptography, especially pairing-based cryptography.

We suggest, as further research, a comparison between our proposed key-exchange protocol and other existing key-exchange protocols [5, 6, 9, 11, 12] in terms of security and performance. Another interesting line of research would be to conduct attacks against our protocol. We think it is strong against the known attacks but we advice further research to be done.

References

[1] BARRETO, P., Kim, H., Lynn, B. and Scott, M., *Efficient Algorithms for Pairing-Based Cryptosystems.* Advances in Cryptology - Crypto'2002, Lecture Notes in Computer Science 2442, Springer-Verlag (2002), pp. 354–368.

[2] BARRETO, P., Lynn, B. and Scott, M., *On the selection of Pairing-Friendly Groups.* Available at http://eprint.iacr.org/2003/086.

[3] BONEH, D. and Franklin, M., *Identity Based Encryption from the Weil Pairing.* Advances in Cryptology - CRYPTO'2001, Springer-Verlag, LNCS 2139, pp. 213-229, 2001.

[4] BORISOV, N., Golberg, I. and Wagner D., *Intercepting Mobile Communications: The Insecurity of 802.11.* Available at http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf.

[5] CHEN, L. and Kudla, C., *Identity Based Authenticated Key Agreement from Pairings.* IACR eprint, report 2002/184.

[6] GUNTHER, C.G., *An Identity-Based Key-Exchange Protocol.* In Proceedings of Eurocrypt 1989, Lecture Notes in Computer Science, Springer-Verlag, pp 29-37, 1989.

[7] HESS, F., *Efficient Identity Based Signature Schemes Based on Pairings.* In: Lecture Notes in Computer Science. Springer-Verlag, 2002. v. 2595, p. 310–324.

[8] SAKAI, R., Ohgishi, K. and Kasahara, M. *Cryptosystems based on Pairing.* Symposium on Cryptography and Information Security, Okinawa, Japan, January 26-28, 2000.

[9] SCOTT, M., *Authenticated ID-based Key Exchange and Remote Log-in with Insecure Token and PIN Number.*IACR eprint, report 2002/164.

[10] SHAMIR, A., *Identity Based Cryptosystems and Signature Schemes.* Advances in Cryptology - CRYPTO'84, Springer-Verlag, LNCS 196, pp. 47-53, 1985.

[11] SMART, N. P., *An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing.* Electronics Letters, Vol 38, pp 630-632, 2002.

[12] ZHANG, F., Liu, S. and Kim, K. *ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings.* IACR eprint, report 2002/122.

SCIS 2004 The 2004 Symposium on Cryptography and Information Security
Sendai, Japan, Jan.27-30, 2004
The Institute of Electronics, Information and Communication Engineers

Pairing-based Problems and Their Applications

Takeshi Okamoto * Tsuyoshi Takagi † Eiji Okamoto *

Abstract— We propose some problems based on bilinear pairing such as Tate pairing over the super singular curve. Our problems consist of five variations and are designed to realize fast computation. So far, many cryptographic schemes have been proposed using bilinear pairing. However, our problem are different from those problems. We also construct three-move type signature schemes using our problems. By analyzing our schemes, we conclude that our schemes obtain good efficiency in some areas. We hope that our problems will give an efficient cryptographic primitive.

Keywords: Tate pairing, Elliptic curve, Pairing-based problem, Three-move type signature

1 Introduction

The pairing-based cryptosystem is a rich area in cryptographic community. Recently, many cryptographic schemes based on the pairings have been developed. For example, there exist the following applications:

- Short signature [BLS02],
- Aggregate signature (multi signature) [BGLS03],
- Tripartite Diffie-Hellman key-exchange [Jou00],
- Identity-based cryptosystem [SOM00, BF01].

The authors think there exists two reasons why this area is developing at such a pace. One reason is that the pairing itself has the special property: the mapping is bilinear. Another reason is that the security of those schemes also depends on the specific problem: discrete logarithm problem in elliptic curve (ECDLP). In this system, the addition operation in an elliptic curve corresponds to modular multiplication in common public-key cryptosystems.

We now focus on the problem which is used in cryptosystem. We take the example of the RSA problem (RSA) which is the most used problem worldwide [RSA78]. Nowadays, there are several variations based on RSA. Note that those problems gives their own useful features. We show some examples:

- The multi-prime RSA problem [Sil00] realizes fast computation,
- The strong RSA problem [BP97] is suitable for group signature cryptosystem [CV91],
- The approximate e-th root problem (AERP) realizes fast signature generation such as ESIGN [OFM98].

In this paper, we propose pairing-based problem. Our main goal is as follows. In the same way as the RSA variant problems, we propose some variations which are pairing-based problems. Consequently, we would like to give desirable contributions in crypto community.

There are five variations named RFP, LFP, SPP, SPUP and SPSP, in this paper. The remarkable point of our problems is that our problems are based on self-exponent pairing although the conventional pairing-based problems depend on ECDLP.

In previous work, Joux proposed some related problems [Jou02]. To define more strictly, we divide his problem into two ones, i.e., RFP and LFP. Now there is no known reducibility between them. We explain that why such a problem is difficult to solve and why we can not give a solution of reducibility. On the other hand, the definition of SPP (resp. SPUP and resp. SPSP) is the first attempt, to the best of author's knowledge. SPP is actually a pairing-based self-powering problem. SPUP is an inverse type of SPSP, i.e, the given part of SPUP is a find part. In the same way, the find part of SPUP is a given part of SPSP.

To improve our problems, we design our problems to realize fast computation. Now there are actually two functions of the pairing: Weil pairing and Tate pairing. Weil pairing can be constructed using Tate pairing and takes large amount of work compared to Tate pairing. For further information, see [GHS02]. Hence in this paper, we explain only Tate pairing for convenience.

Finally, we construct some cryptographic schemes and estimate our schemes. In this paper, we use three signatures which derived from three-move interactive identification schemes like [Sch90]. Our goal in this phase is to make several variations of the signature schemes and evaluate them in terms of the security, the computational efficiency and so on. Each parameter of our scheme is either an element in G_1 or in G_2, where G_1 is an addictive group and G_2 is a multiplicative group. By taking this evaluation, we confirm that our schemes give good efficiency in some cases. We strongly believe that many efficient schemes will be discovered in cryptographic applications.

* University of Tsukuba, Institute of Information Sciences and Electronics, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan, {ken, okamoto}@is.tsukuba.ac.jp

† Technische Universität Darmstadt, Fachbereich Informatik, Alexanderstr.10, D-64283 Darmstadt GERMANY, takagi@informatik.tu-darmstadt.de

This paper is organized as follows. In Section 2, we construct well-designed Tate pairing and propose some problems using such a pairing. We also evaluate the computational efficiency of our problems. In Section 3, we propose some signature schemes which are derived from three-move identification scheme. The security of our schemes is based on our problems. In Section 4, we evaluate the performance of our schemes by comparing them with each other, including ECDSA. The conclusion is given in Section 5.

2 Tate Pairing

In this section we define the Tate pairing. Although there are several different variants of Tate pairing, we define it as suitable for the cryptographic implementation. Indeed, we follow the definition from paper [BKLS02].

Let p be a prime number larger than 3. Let E be an elliptic curve over finite field $\mathbb{F}_p$ defined by $y^2 = x^3 + ax + b$. We choose prime number l such that $gcd(l,p) = 1$. Let k be the smallest integer that satisfies $l|(p^k - 1)$ and $l|\#E(\mathbb{F}_p)$. Here f_P is a function whose divisor $div(f)$ is equal to $l(P) - l(\mathcal{O})$, where P is a point in $E(\mathbb{F}_{p^k})$ and $\mathcal{O}$ is the point of infinity. The Tate pairing is a function

$$(\cdot,\cdot) : E(\mathbb{F}_p)[l] \times E(\mathbb{F}_{p^k})[l] \to \mathbb{F}^*_{p^k},$$

which is defined by

$$(P,Q) = (f_P(Q))^{(p^k-1)/l}, \tag{1}$$

where $P \in E(\mathbb{F}_p)[l]$ and $Q \in E(\mathbb{F}_{p^k})[l]$. The most important property of Tate pairing is the following bilinearlity: $(nP,Q) = (P,nQ) = (P,Q)^n$ for all $n \in \mathbb{Z}$. Note that Q is a point of elliptic curve over extension field $\mathbb{F}_{p^k}$ and two points P, Q are linearly independent, namely $(P,P) \neq 1$.

In the following we treat the Tate pairing constructed on the super singular curve $E : y^2 = x^3 + ax$ over $\mathbb{F}_p$, where $p = 3 \bmod 4$ and some $a \in \mathbb{F}_p$. The MOV degree of the curve is 2, namely $k = 2$. The order of E is equal to $p + 1$, and thus we have $l|(p+1)$ and $l \not| (p-1)$. The points of $E(\mathbb{F}_{p^2})[l]$ can be represented using the distorsion map, namely $\Psi(Q) = (-x, iy)$ where $i \in \mathbb{F}_{p^2}, i^2 = -1$ and $Q = (x,y) \in E(\mathbb{F}_p)$. Barreto et al. pointed out that the scalar value in $\mathbb{F}_p$ can be discarded if $l \not| (q-1)$, because the final multiplication of the Tate pairing computes the exponent $(p^2-1)/l$ in $\mathbb{F}_{p^2}$ [BKLS02]. The Tate pairing over the super singular curve is computed by the following Miller's algorithm. Let n be the bit-length of prime l and let $l[i]$ be the i-th bit of l.

We name Step 3 and Step 4 TDBL and TADD, respectively. Step 5 is the final exponentiation of the Tate pairing. Let $T = (x_1,y_1), P = (x_2,y_2), T + P = (x_3,y_3)$, and $2T = (x_4,y_4)$ with $T \neq \mathcal{O}$, $P \neq \mathcal{O}$, and $T \neq \pm P$. Then ECDBL and ECADD are defined by

$$\begin{aligned}(x_4,y_4) &= (\lambda^2_{dbl}(T) - 2x_1, \lambda_{dbl}(x_1 - x_3) - y_1) \quad (2)\\ (x_3,y_3) &= (\lambda^2_{add}(T,P) - x_1 - x_2,\\ &\qquad \lambda_{add}(x_1 - x_3) - y_1), \quad (3)\end{aligned}$$

Table 1: Computation of Tate pairing

```
INPUT $l, p, k, n$, $P \in E(\mathbb{F}_p)[l]$, and $Q \in E(\mathbb{F}_{p^k})[l]$
OUTPUT $(P,Q)$
1: $T = P, f = 1 \in \mathbb{F}_{p^2}$
2: for $i = n-2$ to 0
3:     $T = ECDBL(T)$, $f = f^2 l_1^{dbl}(T,Q)$
4:     if $l[i] = 1$ then
           $T = ECADD(T,P)$, $f = f l_1^{add}(T,P,Q)$
5: $f = f^{(p^2-1)/l} \in \mathbb{F}_{p^2}$
6: return $f$
```

where

$$\lambda_{dbl}(T) = (3x_1^2 + a)/(2y_1)$$

and

$$\lambda_{add}(T,P) = (y_1 - y_2)/(x_1 - x_2).$$

The lines l_1^{dbl} and l_2^{dbl} are defined as follows:

$$l_1^{dbl}(T,Q) : y - x_1 - \lambda_{dbl}(T)(x_1)(x - x_1), \tag{4}$$
$$l_1^{add}(T,P,Q) : y - x_1 - \lambda_{add}(T,P)(x_1)(x - x_1). \tag{5}$$

Note that the coefficient of lines l_1^{dbl} and l_1^{add} are in $\mathbb{F}_p$, but the value $l_1^{dbl}(T,Q)$ and $l_1^{add}(T,P,Q)$ for $\Psi(Q) = (-x, iy)$ are elements in extension field $\mathbb{F}_{p^2}$. Therefore the updating of $f = f^2 l_1^{dbl}(T,Q)$ and $f = f l_1^{add}(T,P,Q)$ is computed in extension field $\mathbb{F}_{p^2}$.

2.1 Self-Exponent Pairing

In the following we consider the Tate pairing over the super singular curve $E(\mathbb{F}_p) : y^2 = x^3 + ax$. Let $G_l = \{x^{(p^2-1)/l} | x \in \mathbb{F}^*_{p^2}\}$. In this section we deal with the self pairing problem, namely $(S, \Psi(S))$ for $S \in E(\mathbb{F}_p)[l]$. Indeed, we consider the following problem.

Self-Powering pairing Problem (SPP): For a given element $v \in G_l$, find the point $S \in E(\mathbb{F}_p)[l]$ such that $(S, \Psi(S)) = v$.

For a given v there are only two different S and $-S$. Consequently, the following theorem can be proven.

Theorem 2.1 *For any element $v \in G_l$, there is a unique element $S \in E(\mathbb{F}_p)[l]$ up to the sign such that $(S, \Psi(S)) = v$.*

Proof. We assume that there are two different $S_1, S_2 \in E(\mathbb{F}_p)[l]$ such that $(S_1, \Psi(S_1)) = (S_2, \Psi(S_2)) = v$. There exist an integer $1 < c < l$ that satisfies $S_1 = cS_2$. Then we obtain $c^2 = 1 \bmod ord(v)$ due to $v = (S_1, \Psi(S_1)) = (cS_2, \Psi(cS_2)) = (S_2, \Psi(S_2))^{c^2} = v^{c^2}$. The order of G_l is l and thus $ord(v) = 1$ or l. If $ord(v) = 1$ holds, we have $c = 1$ and the theorem is true. Otherwise $c^2 = 1 \bmod l$ has two solutions $c = 1$ and $c = l - 1$. Consequently, the theorem is true.

Additionally, we propose the following problems which are related to SPP.

Self-Powering United pairing Problem (SPUP): For a given element $r, s \in G_l$, find an element $v \in G_l$ such that $(R, \Psi(R)) = r$, $(S, \Psi(S)) = s$, $(R, \Psi(S)) = v$ and $R, S \in E(\mathbb{F}_p)[l]$.

Self-Powering Separated pairing Problem (SPSP): For a given element $v \in G_l$, find an element $r, s \in G_l$ such that $(R, \Psi(R)) = r$, $(S, \Psi(S)) = s$, $(R, \Psi(S)) = v$ and $R, S \in E(\mathbb{F}_p)[l]$.

We now discuss the difficulty of the SPP. Joux introduced the variant of the SPP [Jou02]. To investigate more rigorously, in this paper, we classified the problem into the following two problems:

Right-Fixed pairing Problem (RFP): For a given $v \in G_l$ and a fixed $Q \in E(\mathbb{F}_p)[l]$, find the element $P \in E(\mathbb{F}_p)[l]$ such that $(P, \Psi(Q)) = v$.

Left-Fixed pairing Problem (LFP): For a given $v \in G_l$ and a fixed $P \in E(\mathbb{F}_p)[l]$, find the element $Q \in E(\mathbb{F}_p)[l]$ such that $(P, \Psi(Q)) = v$.

We discuss the difficulty of these problems in the following. The Miller's algorithm computes a sequence of exponent of P based on prime l. We denote them by $P_0 = P, P_1, P_2, ..., P_{k-1}, P_k = \mathcal{O}$, where k is the sum of the bit-length and the hamming weight of l. These points $P_i = (x_i, y_i)$ are used for computing the coefficient of lines l_1^{dbl}, l_2^{dbl} by means of equations (2) and (3). If $\Psi(Q) = (-x, yi)$ is fixed, we can generate an equation with variables x_i, y_i for $i = 0, 1, .., k-1$ due to equations (4) and (5). The variables x_i, y_i are represented by division polynomials of base point $P_0 = (x_0, y_0)$ [Sil86]. Thus for given v and fixed $\Psi(Q)$, we obtain a polynomial whose solution is the point P_0. However the degree of this polynomial is exponentially large at least l^2, and thus we can not solve it in a polynomial time. If $P = (x, y)$ is fixed, we can determine all points P_i for $i = 0, 1, ..., k-1$ using equations (2) and (3), and then all coefficients of lines l_1^{dbl}, l_2^{dbl} can be obtained from equations (4) and (5). Note that each line equation of a linear forms in variable x and y. When we update the accumulator f in TCDBL we have to compute squaring of f. In this case the degree in x, y increases twice, and thus we have to solve an equation over $\mathbb{F}_p$ with exponentially large degree, in order to find $P = (x, y)$. Finally, the SPP is a harder problem, because we have to solve a multivariable polynomial in (x_i, y_i) with exponentially large degree.

2.2 Efficiency Estimation

We estimate the efficiency of computing the Tate pairing over the super singular curve. We assume that parameters p, l are randomly chosen and let b_p, b_l be the bit length of p, l, respectively. The estimation follows the paper [IT02], namely we use the simplified Chudonovsky Jacobian coordinate. Let M be one multiplication cost of $\mathbb{F}_p$. One multiplication and squaring of $\mathbb{F}_p$ can be implemented with $4M$ and $2M$. Izu and Takagi estimated that TDBL and TADD requires $22M$ and $21M$, respectively. The total estimation for evaluating the Tate pairing without final exponentiation requires $32.5 b_l M$, for example, $5200M$ for $b_l = 160$. The final exponentiation with exponent $(p^2-1)/l$ requires $4(2b_p - b_l)M$ using a standard binary method. Therefore, the total running time is $(8b_p + 28.5b_l)M$, for example $8656M$ for $b_p = 512, b_l = 160$. The computation (P, aQ) for $0 < a < l$ additionally requires the computation of aQ, namely $20.5b_l M$ and $3280M$ for $b_l = 160$. The computation $(P, Q)^a$ for $0 < a < l$ additionally requires $4b_l M$ and $640M$ for $b_l = 160$.

Table 2: Variations of the pairing problem.

Problem	Given part	Find part	Equation
RFP	Q, v	P	$v = (P, \Psi(Q))$
LFP	P, v	Q	$v = (P, \Psi(Q))$
SPP	v	S	$v = (S, \Psi(S))$
SPUP	r, s	v	$r = (R, \Psi(R))$ $s = (S, \Psi(S))$ $v = (R, \Psi(S))$
SPSP	v	r, s	$r = (R, \Psi(R))$ $s = (S, \Psi(S))$ $v = (R, \Psi(S))$

Table 3: Comparison of several computation.

Pairing	Amount of work	
(P, Q)	$(8b_p + 28.5b_l)M$	$8656M$
(aP, Q)	$(8b_p + 49b_l)M$	$1,1936M$
$(P, Q)^a$	$(8b_p + 32.5b_l)M$	$9296M$

3 Some Applications

In this section we construct some cryptosystems whose underlying problem are our proposed ones. We propose three-move type signature schemes such as [Sch90].

Let G_1 be an additive group whose order is a prime q and G_2 be a multiplicative of the same order q. $(\cdot, \cdot)$ is a function

$$(\cdot, \cdot) : G_1 \times G_1 \to G_2,$$

which satisfying the bilinear pairing described in Section 2. For simplicity of our protocols, small letter is an element in G_1 and capital letter in G_2 unless explicitly defined in another way. Finally, we define the following cryptographic hash functions.

$$\mathcal{H}_1 : \{0,1\}^* \to \mathbb{F}_q,$$
$$\mathcal{H}_2 : \{0,1\}^* \to G_1.$$

3.1 Scheme 1

Key generation:

- Pick up two elements P, S.
- Compute $v = (P, S)$.

Table 4: Structure of our schemes.

Scheme	Problem	Public-key	Sec. -key	Commitment*1	Challenge*1	Responce*1	Verification
Scheme 1	RFP LFP	P $v=(P,S)$	S	$x=v^r$	$e=\mathcal{H}_1(x,m)$	$Y=(r+e)S$	$x'=(P,Y)v^{-e}$ $e \overset{?}{=} \mathcal{H}_1(x',m)$
Scheme 2	RFP LFP	P $V=-sP$	s	$x=(P,R)$	$E=\mathcal{H}_2(x,m)$	$Y=R+sE$	$x'=(P,Y)(V,E)$ $E \overset{?}{=} \mathcal{H}_2(x',m)$
Scheme 3	SPP	$v=(S,S)$	S	$x=(R,R)$ $\chi=(R,S)^2$	$e=\mathcal{H}_1(x,\chi,m)$	$Y=R+eS$	$x'=(v^e\chi)^{-e}(Y,Y)$ $e \overset{?}{=} \mathcal{H}_1(x',m)$

Note: In the above table, the column of *1 represents a parameter related to the three-move type identification.

Public-key/Secret-key: Signer's public-key is (P,v) and secret-key is S.

Signature Generation:
- Pick up a random number $r \in \mathbb{F}_q$.
- Compute

$$\begin{cases} x = v^r, \\ e = \mathcal{H}(x,m), \\ Y = (r+e)S. \end{cases}$$

Signature: The signature for a message m is (e,Y).

Verification:
- Compute $x' = (P,Y)v^{-e}$.
- Check whether $e = \mathcal{H}_1(x',m)$ holds or not.

This equation holds since

$$\begin{aligned}(P,Y)v^{-e} &= (P,(r+e)S)v^{-e}\\ &= v^{r+e}v^{-e}\\ &= v^r\\ &= x.\end{aligned}$$

3.2 Scheme 2

Key generation:
- Pick up an element P and a random number $s \in \mathbb{F}_q$.
- Compute $V = -sP$.

Public-key/Secret-key: Signer's public-key is P, V and secret-key is s.

Signature Generation:
- Pick up a random element R.
- Compute

$$\begin{cases} x = (P,R), \\ E = \mathcal{H}(x,m), \\ Y = R+sE. \end{cases}$$

Signature: The signature for a message m is (E,Y).

Verification:
- Compute $x' = (P,Y)(V,E)$.
- Check whether $e = \mathcal{H}_1(x',m)$ holds or not.

This equation holds since

$$\begin{aligned}(P,Y)(V,E) &= (P,R+sE)(-sP,E)\\ &= (P,R)(P,sE)(P,E)^{-s}\\ &= (P,R)(P,E)^s(P,E)^{-s}\\ &= (P,R)\\ &= x.\end{aligned}$$

3.3 Scheme 3

Key generation:
- Pick up an element S.
- Compute $v = (S,S)$.

Public-key/Secret-key: Signer's public-key is v and secret-key is S.

Signature Generation:
- Pick up a random element R.
- Compute

$$\begin{cases} x = (R,R), \chi = (R,S)^2, \\ e = \mathcal{H}(x,\chi,m), \\ Y = R+eS. \end{cases}$$

Signature: The signature for a message m is (χ,e,Y).

Verification:
- Compute $x' = (v^e\chi)^{-e}(Y,Y)$.
- Check whether $e = \mathcal{H}(x',m)$ holds or not.

This equation holds since

$$\begin{aligned}(v^e\chi)^{-e}(Y,Y) &= (v^e\chi)^{-e}(R+eS,R+eS)\\ &= v^{-e^2}\chi^{-e}(R,R)(R,eS)(eS,R)(eS,eS)\\ &= v^{-e^2}\chi^{-e}(S,S)^{e^2}((R,S)^2)^e(R,R)\\ &= v^{-e^2}\chi^{-e}v^{e^2}\chi^e x\\ &= x.\end{aligned}$$

4 Discussions

In this section we analyze our schemes. Table 4 gives the structure of our schemes.

On Scheme 1

Reducibility: It is not known at present to hold that RFP reduces to ($\leq$) LFP and LFP $\leq$ RFP. Those reductions remain open problem. This means that even if an adversary knows the algorithm to solve RFP, she can not break Scheme 1, because Scheme 1 is LFP-based scheme. If we modify the protocol of Scheme 1 by changing $v = (P, S)$, $x' = (P, Y)v^{-e}$ into $v = (S, P)$, $x' = (Y, P)v^{-e}$, such a modified scheme based on RFP-based one. In this case, the adversary can break this modified scheme. The same argument holds in case Scheme 2.

Fast on-line computation: In Scheme 1, $Y = (r + e)S$. Note that when we calculate the value of Y, we first compute $(r + e) \in \mathbb{Z}$. This means that neither multiplication nor modular reduction is used in this phase. Moreover, S is a fixed point since S is a secret-key. Consequently, one may say that Scheme 1 requires fast signature generation in the on-line phase compared to Scheme 1, 2 and ECDSA.

On Scheme 2

Hashing: In Scheme 2, we use the hash function $\mathcal{H}_2$ such that $\mathcal{H}_2 : \{0,1\}^* \to G_1$. In case of implementation, we must take care of the designing of such hash functions. We need the structure to relax the hashing requirement. For more information, we refer to [BLS02].

Fast verification: As for the Scheme 2, the actual computation in verification is $x' = (P, V)(V, E)$. Note that there is no need to compute aQ, (P, aQ) or $(P, Q)^a$ for some P, Q and $a \in \mathbb{F}_a$. On the other hand, Scheme 1, 3 and ECDSA need such computations. Based on Table 3, one may conclude that Scheme 2 realizes the fast verification compared to the other schemes.

On Scheme 3:

Self-exponent: It is not known that SPP $\leq$ ECDLP and ECDLP $\leq$ SPP. We next discuss the reducibility on SPUP and SPSP. As a result, SPUP $\leq$ SPP. However, it is not known whether SPP $\leq$ SPUP. Finally, we write the following open problem. It is not known both SPP $\leq$ SPSP and SPSP $\leq$ SPP.

Security: In verification, e is used as an exponent since $x' = (v^e\chi)^{-e}(Y, Y)$. This means that an adversary can break Scheme 3 using the algorithm to solve discrete logarithm problem in finite field (DLP). If an adversary knows the algorithm to solve SPUP, then she can compute χ using v and x. However, we can not describe the algorithm to break Scheme 3 using the algorithm to solve SPUP.

5 Conclusion

In this paper, we have proposed some pairing-based problems named RFP, LFP, SPP, SPUP and SPSP. We also have shown that those problems can be efficient cryptographic primitives by presenting three signature schemes named Scheme 1, 2, and 3.

As a result, Scheme 1 realizes fast signature generation in the on-line phase. Scheme 2 realizes fast verification thanks to a unique equation in verification. Scheme 3 may be more secure than ECDSA. Now there are many open problems with respect to our problems.

Due to the lack of space, we have just focused on the signature schemes as application. Hence the other cryptosystem, such as encryption and key distribution also should be considered. We believe that our problems and schemes would contribute to the cryptographic community.

References

[BKLS02] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," CRYPTO 2002, LNCS 2442, pp.354-368, 2002.

[BP97] M. Bellare and P. Rogaway, "Collision-resistant hashing: towards practical," CRYPTO 1997, LNCS 1294, pp. 470-484, 1997.

[BF01] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," CRYPTO 2001, LNCS 2139, pp. 213-229, 2001.

[BLS02] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," ASIACRYPT 2001, LNCS 2248, pp.514-532, 2002.

[BGLS03] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," EUROCRYPT 2003, LNCS 2656, pp. , 2003.

[CV91] D. Chaum and E. van Heijst, "Group signatures," EUROCRYPT 1991, LNCS 547, pp. 257-265," Springer-Verlag, 1991.

[GHS02] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate Pairing," ANTS V, LNCS 2369, pp.324-337, 2002.

[Hes02] F. Hess, "Exponent Group Signature Schemes and Efficient Identity Based Signature Schemes Based on Pairings," Cryptology ePrint Archive, Report 2002/012, available at `http://eprint.iacr.org/2002/012/`.

[IT02] T. Izu and T. Takagi, "Efficient Computations of the Tate Pairing for the Large MOV Degrees," ICISC 2002, LNCS 2587, pp.283-297, 2002.

[Jou00] A. Joux, "A one-round protocol for tripartite Diffie-Hellman," Algorithm Number Theory Symposium - ANTS IV, LNCS 1838, pp. 385-394, 2000.

[Jou02] A. Joux, "The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems (survey)," ANTS V, LNCS 2369, pp.20-32, 2002.

[OFM98] T. Okamoto, E. Fujisaki and H. Morita. TSH-ESIGN, "Efficient Digital Signature Scheme Using Trisection Size Hash," Submission to P1363a, 1998.

[RSA78] R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital cryptosystems," Communications of the ACM, 21(2): pp.120-126, 1978.

[SOM00] R. Sakai, K. Ohgishi and M. Kasahara, "Cryptosystems based on pairing," Symposium on Cryptography and Information Security (SCIS2000), Okinawa, Japan, 2000.

[Sch90] C.P. Schnorr. "Efficient identification and signatures for smart cards," CRYPTO 1989, LNCS 435, pages 239-251, 1990.

[Sil86] J. Silverman, "The Arithmetic of Elliptic Curves," GMT 106, Springer-Verlag, 1986.

[Sil00] R. Silverman, "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths," RSA Laboratories Bulletin No. 13, April 2000, Available at `http:www.rsasecurity.com/rsalabs/`.

PAPER *Special Section on Discrete Mathematics and Its Applications*

A Fast Signature Scheme with New On-line Computation

Takeshi OKAMOTO[†a)], *Regular Member*, Hirofumi KATSUNO[††b)], *Nonmember*, *and* Eiji OKAMOTO[†c)], *Regular Member*

SUMMARY In this paper, we propose a fast signature scheme which realizes short transmissions and minimal on-line computation. Our scheme requires a modular exponentiation as preprocessing (i.e., off-line computation). However, we need to acknowledge the existance of the following remarkable properties: neither multiplication nor modular reduction is used in the actual signature generation (i.e., on-line computation). Our scheme requires only two operations: hashing and addition. Although some fast signature schemes with small on-line computation have been proposed so far, those schemes require multiplication or modular reduction in the on-line phase. This leads to a large amount of work compared to that of addition. As far as we know, this is the first approach to obtain the fast signature without those two calculus methods.

key words: *digital signature, on-line computation, random oracle model, provable security*

1. Introduction

1.1 Motivation

Nowadays, a signature scheme is an important tool for secure communication over open network. Consequently, there is a strong need for more *compact* signature schemes to spread public-key infrastructure (PKI) system. In this case, the compactness means the efficiency of both computational work and transmitted data size. Such compactness is more convenient for users and is more adaptable for various applications.

We are now going to have a look into the computational efficiency in signature schemes and focusing on the signer's computational work which can be derived from a three-pass identification scheme like [23]. In such a signature scheme, we can see two kinds of computation for the signer: *pre-computation* and *(actual) signature generation.*

The pre-computation can compute values without a message to be signed, and can be executed during idle time and completely independent of the message. This means that such a computational cost does not influence the real-time computing. We are defining such a computation as *off-line* processing. On the other hand, the (actual) signature generation does directly influence the processing time because a message is indispensable for computing. We are defining that such a computation as *on-line* processing.

To estimate the efficiency of a signature scheme, we must separately consider the two types of computation. Needless to say, the fast signature scheme with the on-line computational efficiency, can make digital signatures much more practical in a variety of scenarios.

Let us now focus on the various operations which appear in the on-line signature generation. In the case of usual signature, there are four kinds of operations: multiplication, modular reduction, addition and hashing. Although multiplication or modular reduction require a large amount of work, the computational work for addition as well as hashing is quite small. Consequently, the most desirable way for fast on-line computing is to eliminate both of such heavy operations.

We now want to emphasize that our signature system in this paper is used under the following specifications:

- Some values which are independent of a message, are always computed in the off-line phase.
- At the time of signature creation, only on-line processing is performed.

Because of the above setting, if a signer wants to realize a fast signature generation, she needs to reduce the amount of work in the on-line phase. As a result, our scheme is quite useful under the following circumstances:

1. In case the real-time processing is required for a long time: For example, suppose a user is dealing with a streaming media. To avoid the deviation of copyright, it is desirable for the server to sign the contents using a secret-key which is different for every user. To realize a streaming implementation, it is necessary to create a fast on-line signature.
2. In case many signatures should be generated simultaneously: In electronic authentication systems, so many users access to a center using on-line service. In this case, the center creates signatures of messages which are selected by the users, and transmits those to the users. When the center needs to generate many signatures simultaneously, high-

Manuscript received August 25, 2003.

[†]The authors are with the Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan.

[††]The authors are with the Department of Information Science, Tokyo Denki University, Hatoyama-machi, Hiki-gun, Saitama, 350-0394, Japan.

a) E-mail: ken@is.tsukuba.ac.jp

b) E-mail: katsuno@j.dendai.ac.jp

c) E-mail: okamoto@is.tsukuba.ac.jp

speed signature generation is required.

3. In case application is required: There are some applications such that a center picks up some values in the off-line phase. Usually, those values consists of a random number and a calculation result. A user uses those values as a *coupon*. By using such a coupon, a user signs the message by performing on-line processing. For example, one system was proposed in [12].

1.2 Related Work

To realize the fast on-line signature generation, Girault [6] has modified Schnorr's scheme [23] which uses an RSA-modulus (i.e., the modulus is a product of two distinct primes) instead of a prime modulus. This modification leads to no modulo reduction in the on-line signature generation. Therefore, Girault's scheme creates faster processing for signature generation compared to Schnorr's one. In 1998, Poupard and Stern [20] investigated and gave provable security for Girault's scheme, and named that scheme GPS. According to [20], we say that a signature scheme, in which modulo reduction is not used in the on-line signature generation, is *on the fly* scheme.

In 1999, Poupard and Stern [21] proposed another on the fly scheme (PS), whose security relies on the difficulty of integer factoring. In 2001, Okamoto, Tada and Miyaji [15] proposed on the fly scheme (OTM1) by improving PS in the computational work and transmitted data size.

In 2002, Okamoto, Tada and Miyaji [16] proposed another signature (OTM2) in which no multiplication is used in the on-line phase. This means that the phase involves only hashing, addition and modular reduction.

1.3 Our Contribution

In this paper, we propose a fast signature scheme which is derived from a three-pass identification scheme. This paper is analyzed in the outline of [14]. Our scheme has small key-storage inefficiencies such as the size of public-key or that of secret-key, but realizes quite fast on-line signature generation.

Our approach for reducing the on-line computation is different from that of previous work. That is, both modular reduction and multiplication are eliminated in our scheme, although the previous schemes can eliminate only one operation: either modular reduction or multiplication. As a result, our scheme consists of only hashing and addition in the on-line phase. Until now, some fast signature schemes which have the property of small on-line computation, have been proposed. Consequently, in the on-line computation, multiplication is used in on the fly schemes and the modular reduction is used in the OTM2 scheme. We emphasize that those two operations are not used in our scheme.

Let us now compare the computational efficiency in the multiplication, the modular reduction and the addition. In the multiplication, a recursive algorithm due to [8] reduces the complexity of the multiplying. In the modular reduction, we can use the efficient methods such as [1], [11]. Nevertheless, those methods suffer from a large amount of work compared to the addition. On the other hand, addition is computed using a more straightforward way, and reduces the considerable complexity in the on-line phase.

Consequently, the addition in our scheme is faster than the other operations from implementation point of view. For more detailed analysis of the operations, see [9].

We now give a concrete evaluation. Compared with Feige-Fiat-Shamir [4] (FFS), GPS, PS, OTM1 and OTM2, the size of a signature in our scheme can be reduced by at least 70%, 74%, 56%, 17%, 32%, respectively.

As for the security, our scheme is as secure as integer factoring problem based on an RSA modulus n (in the random oracle model). To satisfy the security, our scheme uses a public key g with specific structure, called *asymmetric basis*, which is a variant of [17]. This leads to good efficiency in terms of both transmitted data size (i.e., the size of signature) and amount of work (in the on-line phase).

1.4 Outline of Our Paper

This paper is organized as follows. In Section 2, we give the overview of fast signature schemes. In Section 3, we introduce our signature scheme. In Section 4, we show that our scheme is provably secure under the assumption that factoring problem is computationally hard to solve. In Section 5, we suggest practical values of our signature scheme and describe the implementation notes. In Section 6, we evaluate the performance of our signature scheme by comparing it with existing fast signature schemes: FFS, GPS, PS, OTM1, OTM2. The conclusion is given in Section 7.

2. Previous Schemes

In this section, we survey the previous work.

We first introduce some notations. $\varphi(\cdot)$ denotes Euler totient function, i.e., $\varphi(n)$ is the number of the natural numbers less than n and coprime to n. $\lambda(\cdot)$ denotes so-called Carmichael function, i.e, $\lambda(n)$ is the greatest number among the possible orders of elements in $\mathbf{Z}_n^*$. Let $m \in \{0,1\}^*$ be a message to be signed. Let $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ be a hash function. The right side of $\mathcal{H}(x) = (e_1, e_2, \cdots, e_k)$ means the bit strings of bit length k, where $x \in \{0,1\}^*$ and $e_i \in \{0,1\}$ $(1 \le i \le k)$. The order of an element $g \in \mathbf{Z}_n^*$ is represented as $\mathrm{Ord}_n(g)$. $|x|$ denotes the binary length for a positive integer x. We say that p is a strong prime if p and p'

are primes satisfying $p = 2p' + 1$.

We now show the overview of the previous schemes. For all schemes, the signature for m is (x, e, y).

FFS [4]: The public key is (n, v_i), where n is an RSA modulus and $v_i = s_i^{-2} \bmod n$ for $1 \le i \le k$. $s_i \in \mathbb{Z}_n$ $(1 \le i \le k)$ is picked up at random. The secret key is s_i $(1 \le i \le k)$. For signature generation, a signer picks up $r \in \mathbb{Z}_n$ at random. She also computes $x = r^2 \bmod n$, $e = \mathcal{H}(x, m)$ and $y = r \cdot \Pi_{i=1}^{k} s_i^{e_i} \bmod n$. A verifier finally checks whether $x = y^2 \cdot \Pi_{i=1}^{k} v_i^{e_i} \bmod n$ holds or not. Each parameter (k, n) satisfies the condition $2^k \ll n$.

GPS [20]: The public key is (n, g, v), where n is an RSA modulus, $g \in \mathbb{Z}_n^*$ is an element with high order, and $v = g^{-s} \bmod n$. $s \in \mathbb{Z}_{2^k}$ is picked up at random. The secret key is s. For signature generation, a signer picks up $r \in A$ at random. She also computes $x = g^r \bmod n$, $e = \mathcal{H}(x, m)$ and $y = r + se$ (in $\mathbb{Z}$). For verification, a verifier finally checks the validity whether $x = g^y v^e \bmod n$ holds or not. Each parameter (A, B, k) satisfies $B \ll 2^k \ll A$, where $e \in \mathbb{Z}_B$.

PS [21]: The public key is (n, g), where n is a product of strong primes p and q, and $g \in \mathbb{Z}_n^*$ is an element such that $\mathrm{Ord}_n(g) \in \{\lambda(n), \lambda(n)/2\}$. The secret key is $s = n - \varphi(n)$ $(= p + q + 1)$, where $|s| = k$. For signature generation, a signer picks up $r \in \mathbb{Z}_A$ at random. She also computes $x = g^r \bmod n$, $e = \mathcal{H}(x, m)$ and $y = r + se$ (in $\mathbb{Z}$). For verification, a verifier finally checks whether both $y < A$ and $x = g^{y - ne} \bmod n$ hold or not. Each parameter (A, B, k) satisfies $B \ll 2^k \ll A$, where $e \in \mathbb{Z}_B$.

OTM1 [15]: The public key is (n, g, z), where $n = \prod_{i=1}^{t} p_i$ for an integer $t \ge 2$, $g \in \mathbb{Z}_n^*$ is an element such that $\mathrm{Ord}_n(g) = q$. $z \in \mathbb{Z}_{2^a}$ is picked up at random. The secret key is $s = z \bmod q$, where $|s| = k$. For signature generation, a signer picks up $r \in 2^a$ at random. She also computes $x = g^r \bmod n$, $e \in_R = \mathcal{H}(x, m)$ and $y = r + se$ (in $\mathbb{Z}$). A verifier finally checks whether both $y < 2^{a+1}$ and $x = g^{y - ze}$ hold or not. Each parameter (a, b, c, k) satisfies the condition $2^b \ll 2^k \ll 2^a \ll 2^{bc}$, where $e \in 2^b$.

OTM2 [16]: The public key is (n, g), where $n = pq$ and $g \in \mathbb{Z}_n^*$ is an asymmetric basic described in Section 3. The secret key is a prime number s with $\mathrm{Ord}_n(g) = s$. For signature generation, a signer picks up $r \in 2^a$ at random. She also computes $x = g^r \bmod n$, $e = \mathcal{H}(x, m)$ and $y = r + (e \bmod s)$. The signature for m is (x, e, y). A verifier finally checks whether both $y < 2^a + 2^k$ and $x = g^{y-e} \bmod n$ hold or not. Each parameter (a, b, k, s) satisfies the condition $2^{k-1} \le 2^k \ll 2^a \ll 2^b$, where $e \in 2^b$.

Remark. For each scheme mentioned above, we set the security parameter k by taking the author's definition. Therefore, each scheme is defined such that $k = |n|$, $k = |e|$ or $k = |s|$, so one scheme may be different from another. However, that is not an essential matter because k is a security parameter,.

3. Signature Scheme

In this section, we introduce our signature scheme.

Note that our signature scheme is derived from three-pass interactive zero-knowledge identification scheme. That scheme can be translated into a signature scheme by using the methods of [5], [23]

For reader's convenience, we now briefly show some parameters. Let k, a, s and $\mathcal{O}$ be four integers satisfying $2^k \ll 2^s \ll 2^a$ and $2^k \ll \mathcal{O}$, where (k, a, s) and $\mathcal{O}$ are public and secret parameters, respectively. The more detailed relations between those parameters are analyzed in Section 5.

In our scheme, we use slightly generalized asymmetric basis [17], which is defined as follows.

Definition 3.1 (Asymmetric basis): Let n be an RSA modulus such that $n = pq$. Then we say that g is an asymmetric basis in $\mathbb{Z}_n^*$ if the multiplicity of 2 in $\mathrm{Ord}_p(g)$ is not equal to that of 2 in $\mathrm{Ord}_q(g)$. ■

We hereafter show the signature algorithms of our scheme. The intuitive description is shown in Fig. 1.

Key generation: A signer who wants to generate key parameters, executes the following steps.

1. Pick up two large primes p and q.
2. Compute $n = pq$.
3. Pick up an asymmetric basis $g \in \mathbb{Z}_n^*$, where $\mathrm{Ord}_n(g) = \mathcal{O}$.
4. Pick up k random integers $s_1, s_2, \cdots, s_k \in \mathbb{Z}_{2^s}$.
5. Compute $v_i = g^{-s_i} \bmod n$; each i $(1 \le i \le k)$.

Public-key/Secret-key: Signer's public-key is $(v_1, v_2, \cdots, v_k; n; g)$ and secret-key is $(s_1, s_2, \cdots, s_k)$.

Signature generation: Suppose a signer who has a public-key and the corresponding secret-key, generates the signature of her message m. The signer executes the following steps.

1. Pick up a random number $r \in \mathbb{Z}_{2^a}$.
2. Compute $x = g^r \bmod n$.
3. Compute $e = \mathcal{H}(x, m) = (e_1, e_2, \cdots, e_k)$; each $e_i \in \{0, 1\}$.
4. Compute $y = r + \Sigma_{i=1}^{k} S_i$, where

$$S_i = \begin{cases} 0 & \text{if } e_i = 0; \\ s_i & \text{if } e_i = 1. \end{cases}$$

Remark. As you see, our scheme requires only hashing and addition operations in the on-line phase for signature generation. This means that our scheme satisfies

Key generation

Parameters:

- RSA modulus $n = pq$
- asymmetric basis $g \in Z_n^*$ with $\mathrm{Ord}_n(g) = \mathcal{O}$
- for each i $(1 \le i \le k)$ $[\, s_i \in_R \mathbb{Z}_{2^s},\ \ v_i = g^{-s_i} \bmod n \,]$
- $2^k \ll 2^s \ll 2^a,\ 2^k \ll \mathcal{O}$

Signature generation and verification

$\boxed{\text{Signer}}$ $\boxed{\text{Verifier}}$

Public-key: v_i, n, g
Secret-key: s_i

[Off-line phase]

$r \in_R Z_{2^a}$
$x = g^r \bmod n$

[On-line phase]

$e = \mathcal{H}(x, m)$
$:= (e_1, e_2, \cdots, e_k)$ for each $e_i \in \{0, 1\}$

$y = r + \Sigma_{i=1}^{k} S_i$, where
$$S_i = \begin{cases} 0 & \text{if } e_i = 0; \\ s_i & \text{if } e_i = 1. \end{cases}$$

$$\xrightarrow{m, (x, e, y)}$$

$x' = g^y \, \Pi_{i=1}^{k} v_i^{e_i} \bmod n$
$e \overset{?}{=} \mathcal{H}(x', m)$

Fig. 1 Our signature scheme.

the conditions which are described in Section 1.

Signature: The signature for a message m is (x, e, y).

Verification: Suppose a verifier who has the signer's public key and the corresponding message, checks the validity of the signature for m. The signer executes the following steps.

1. Check whether $0 < y < 2^a + k2^s$ holds or not. If the equation does not hold, then reject the signature and stop this protocol.
2. Compute $e' = \mathcal{H}(x, m)$ and $x' = g^y \, \Pi_{i=1}^{k} v_i^{e_i} \bmod n$
3. Check whether both $e = e'$ and $x = x'$ hold or not. If both equations hold, accept the signature. Otherwise reject it.

4. Security Analysis

In this section, we analyze the security of our scheme.

We say that a signature scheme is *secure*, if no polynomial time adversary can existentially forge a signature in *the adaptive chosen message attack*. In this section, we show that our signatures scheme is secure, by using *the forking lemma* in [18], and showing that a signing oracle can be simulated by a polynomial-time machine in *the random oracle model* [2], which is under the one key attack scenario.

Lemma 4.1: Let $n = pq$ be an RSA modulus. Let g be an asymmetric basis in $\mathbb{Z}_n^*$. If we find a non-negative integer L such that $g^L = 1 \bmod n$, then we can construct a Turing machine M which on input n, g and L outputs a factor of n in time $O(|L||n|^2)$

Proof. To prove this lemma, we first describe a simulation. M executes the following steps.

1. Extract the odd part of L, i.e., compute a and b such that $L = 2^a b$, where $a \in \mathbb{Z}$ and b is an odd number.
2. Compute x by using the following loop algorithm:
```
for ($\ell = 0, 1, \cdots, a-1$) do
    Compute $x = \gcd(g^{2^\ell b} - 1 \bmod n, n)$.
    If $x \neq 1$ then exit this loop.
end for
```
3. Compute $y = n/x$.
4. Output (x, y).

Note that, in Step 4, two integers (x, y) are the factors of n. We now explain why M can generate such factors by using above steps.

In the same way as Step 1, we would like to extract the odd parts of $\mathrm{Ord}_p(g)$ and $\mathrm{Ord}_q(g)$, respectively. We define $(\alpha, \beta, \gamma, \delta)$ such that $\mathrm{Ord}_p(g) = 2^\alpha \beta$ and $\mathrm{Ord}_q(g) = 2^\gamma \delta$, where $\alpha, \gamma \in \mathbb{Z}$ and (β, δ) are odd numbers.

Since g is an asymmetric basis in $\mathbb{Z}_n^*$, $\alpha \neq \gamma$. Hence

$$g^{2^a b} = 1 \bmod n,$$

$$g^{2^a b} = 1 \bmod p \text{ and } g^{2^a b} = 1 \bmod q.$$

Those equations lead to

$$g^{2^{\alpha-1} b} = -1 \bmod p \text{ and } g^{2^{\alpha-1} b} = 1 \bmod q.$$

We assume $\alpha > \gamma$ in this proof. Then

$$p \nmid g^{2^{\alpha-1} b} - 1 \text{ and } q \mid g^{2^{\alpha-1} b} - 1,$$

where b is the same number in Step 1. Hence

$$n \nmid g^{2^{\alpha-1} b} - 1.$$

Consequently, M can find a factor x of n by computing $\gcd(g^{2^\ell b} - 1 \bmod n, n)$ for an integer $0 \le \ell \le a-1$.

Note that the modular exponentiation algorithm has a running time of $O(|L||n|^2)$. In the same way, the extended Euclidean algorithm has that of $O(|n|^2)$. Therefore, M can execute the above steps in time $O(|L||n|^2)$. □

We say that a positive function $f(k) : \mathbb{N} \to \mathbb{R}$ is said to be *negligible*, if for any c, there exists a k_c such that $f(k) \le k^{-c}$ for any $k \ge k_c$. Otherwise f is said to be *non negligible*.

Let Q be the number of queries which a polynomial-time adversary $\mathcal{A}$ (adaptive chosen-message attacker) can ask to the random oracle. Let R be the number of queries which $\mathcal{A}$ can ask to the the actual signer.

$|x|$ denotes the binary length for a positive integer x.

Theorem 4.2: Assume that, in our scheme, $2^s/2^a$ is negligible. Also assume that $\mathcal{A}$ can forge a signature with non-negligible probability $\varepsilon \ge 10(R+1)(R+Q)/2^a$, and with the average running time T. Then we can construct a polynomial time machine M which can solve the factorization of n with non negligible probability in expected time $O(QT/\varepsilon + |L||n|^2)$.k

Proof. Let us first show the outline of this proof. Note that our signature is of the form (x, e, y) and each of (x, e, y) is corresponds to one of the three phases of an identification protocol. If parameters (x, e, y) can be statistically simulated by a polynomial time machine M without knowing the secret-key, then there is an algorithm $\mathcal{A}'$ who produces two valid signatures like (x, e, y) and (x, e', y') with $e \neq e'$. Consequently we can compute two factors of n by using such signatures. For more details see [18].

We now show that our signature can be statistically simulated by a polynomial time machine. We now denote, by $p(\alpha, \beta, \gamma)$ and $p'(\alpha, \beta, \gamma)$, the probabilities that (α, β, γ) is output by the signature algorithm and the simulator, respectively. We set $\phi = k2^s$. Let $\mathcal{R} : \{0,1\}^* \to \{0,1\}^k$ be an ideal hash function (i.e., random oracle) for a given message $m \in \{0,1\}^*$. Here $\mathcal{R}(x_1, x_2) = (\beta_1, \beta_2, \cdots, \beta_k) = \beta$ for two integers x_1 and x_2. For an integer A and a positive constant Δ, $\mathcal{N}(\mathcal{R}, A, \Delta)$ is defined to be the number of pairs (e, y) such that $0 \le e < 2^{k+1}$, $A \le y < A + \Delta$ and $\mathcal{R}(g^y \Pi_{i=1}^k v_i^{e_i} \bmod n, m) = e$.

Let $\wp$ be a predicate. $\chi(\wp)$ is the characteristic function of $\wp$, that is, if $\wp$ is true then $\chi(\wp) = 1$. Otherwise $\chi(\wp) = 0$. Then we have the following:

$$p(\alpha, \beta, \gamma) = \frac{\chi\left(\begin{array}{l} g^\gamma \Pi_{i=1}^k v_i^{\beta_i} \bmod n = \alpha, \\ \mathcal{R}(\alpha, m) = \beta, \\ 0 \le \gamma - \Sigma_{i=1}^k s_i \beta_i < 2^a \end{array}\right)}{2^a}$$

and

$$p'(\alpha, \beta, \gamma) = \frac{\chi\left(\begin{array}{l} g^\gamma \Pi_{i=1}^k v_i^{\beta_i} \bmod n = \alpha, \\ \mathcal{R}(\alpha, m) = \beta, \\ \phi \le \gamma < 2^a - \phi \end{array}\right)}{\mathcal{N}(\mathcal{R}, \phi, 2^a - \phi)}.$$

Therefore, the summation

$$\Sigma = \sum_{\alpha, \beta, \gamma} |p(\alpha, \beta, \gamma) - p'(\alpha, \beta, \gamma)|$$

has an upper bound of $2^s k/2^a$, because $\Sigma = 2(1 - \mathcal{N}(\mathcal{R}, \phi, 2^a - \phi)/2^a)$ holds similarly with [20]. Consequently, the output by real signer and that by the simulator are statistically indistinguishable.

Next, by using the technique in [18], we can get a multiple of $\mathrm{Ord}_n(g)$, L, such that $g^L = 1 \bmod n$. This means that we can obtain factorization of n. Hence this proves the theorem. □

5. Analysis of Our Scheme

In this section, we present the conditions of the parameters to maintain the security of our signature scheme. In addition, we show how to implement the parameters. Moreover, an optimized scheme which reduces the size of signature, is also given.

5.1 Parameter Generation

Parameters a and s: For the security reason, the values of a and s shall satisfy: $a = s + k/2 + \kappa_1$ and $s = k + \kappa_2$, where k is a security parameter and both κ_1 and κ_2 are information leak parameters. a must be satisfied such that $1/2^{\kappa_1}$ is negligible for the security parameter k. In addition, s and $1/2^{\kappa_2}$ have the same conditions. In implementation, we should set κ_1 (resp.

Table 1 Performance of signature schemes.

Scheme	Underlying problem	Off-line comput.*3 ($\times M$)	On-line comput.	Veri. ($\times M$)	Public key (bits)	Secret key (bits)	Sig. (bits)
Our scheme	Fact.*2	61	**addition 40 times (160 bits)**	378	83968	12800	**320**
FFS	Fact.	1	mod. multi. $41M$	42	82944	81920	1104
GPS	Disc. log. (modulo n)	381	multi. 80×1024	1796	3072	1024	1264
PS	Fact.	381	multi. 80×512	1656	1024	513	736
OTM1*1	Fact.*2	61	multi. 80×160	552	2356	160	384
OTM2	Fact.*2	61	mod. reduct. $248 \bmod 160$	372	2048	160	472

Abbreviation:

- M represents the computational cost for one multiplication under a 1024-bit modulus.
- $\gamma \times \delta$ represents the computational cost for multiplication of an γ-bit number and a δ-bit number on $\mathbf{Z}$.
- $\alpha \bmod \beta$ represents the computational cost for modular reduction of an α-bit number and a β-bit number modulus.

Parameters:

- For all schemes, we set $|n| = 1024$ and optimize the size of signature by using the technique in Section 5.2.
- All the values related to the amount of work are shown by the average running time.
- Our scheme has $a = 240$ and $s = 160$ by taking $k = 80$ and $|\mathcal{O}| = 160$.
- FFS has $k = 80$.
- GPS has $|A| = 1184$ and $|B| = 80$ by taking $k = 1024$.
- PS has $|A| = 656$ and $|B| = 80$ by taking $k = 513$.
- OTM1 has $a = 304$, $b = 80$, $c = 288$ and $t = 2$ by taking $k = 160$.
- OTM2 has $a = 224$ and $b = 248$ by taking $k = 160$, $\kappa_1 = 64$ and $\kappa_2 = 24$.

Some Notes:

- For all schemes, we set up the parameter on the scenario of the one-key attack in [21].
- For respective computational cost, a primitive arithmetic of binary methods [9] is used, e.g. amount of work for $g^\alpha \bmod n$ is $\frac{3}{2}|\alpha|M$ if $|n| = 1024$. Of course there exist more sophisticated methods which reduce the amount of computational work. However we think they represent the actual estimate without loss of generosity.
- In the row of "*1",i.e., OTM1, n is an RSA modulus and g is an asymmetric basis in $\mathbf{Z}_n^*$ to keep the same security level as our scheme. Furthermore, the size of public-key is optimized as follows. We regard actual public-key as (n, g), and z is computed by $z = \mathcal{H}'(n, g)$, where $\mathcal{H}'$ is a hash function $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^c$.
- The factoring problem of "*2" is different from *usual* integer factoring problem. This is a problem on input RSA modulus n and the asymmetric basis g, outputs the factor of n.
- In the column of "*3", the signer uses the technique of CRT. In this case, the signer must secretly have the factors of n, p and q.

κ_2) greater than 80 bits. Consequently, $a = 240$ and $s = 160$ are reliable values.

Parameter $\mathcal{O}$: Let us consider the attack where an adversary computes $\mathcal{O}$ only from the information of the public-key (n, g). We can see the algorithms to extract $\mathcal{O}$, such as *Pollard lambda method* in [19] and the *baby-step giant-step method* in [10]. One may say that the former is better than the latter since it has same computational complexity (exponential-time: $O(\sqrt{\mathcal{O}})$) but does not need large memory. The size of $\mathcal{O}$ shall be set up such that the above algorithms cannot be applied for the security parameter k. In implementation, we should set $\mathcal{O}$ greater than 160 bits for the security reason.

Choice of p, q and g: We describe how to find p, q and an asymmetric basis g in $\mathbf{Z}_n^*$.

1. Pick up two primes $p = 2p'p'' + 1$ and $q = 2q'q'' + 1$ such that p' and q' are also primes, and p'' and q'' are odd numbers.
2. Choose $\alpha_p \in \mathbf{Z}_p^*$ satisfying $g_p = \alpha_p^{(p-1)/p'} \neq 1 \bmod$

p. In the same way, choose $\alpha_q \in \mathbb{Z}_q^*$ satisfying $\alpha_q \neq q-1 \bmod q$, $\alpha_q^{(q-1)/2} \neq 1 \bmod q$ and $g_q = \alpha_q^{(q-1)/2q'} \neq 1 \bmod q$.

3. Compute $n = pq$ and $g = q(q^{-1} \bmod p)g_p + p(p^{-1} \bmod q)g_q \bmod n$.

In the last step, g is computed by using the technique of Chinese Reminder Theorem (CRT). Note that $\mathrm{Ord}_p(g) = p'$ and $\mathrm{Ord}_q(g) = 2q'$. Therefore, $\mathrm{Ord}_n(g) = \mathrm{lcm}(p', 2q') = 2p'q'$.

Choice of $\mathcal{H}$: If $\mathcal{H}$ is an *ideal* hash function (i.e., random oracle), then the proposed signature scheme would be *secure* under the meaning of [18]. Since such a random function does not exist in the real world, in implementation, one may recommend to adopt MD5 by [22] or SHA-1 by [13], each of which is designed so that the algorithm can be a collision intractable hash function [3].

5.2 Optimized Scheme

As for the signature scheme in Section 3, we can reduce the size of the signature. Consequently, communication load gets smaller than before. When we have two parameters e and y, the parameter x can be generated by computing $x = g^y \Pi_{i=1}^k v_i^{e_i} \bmod n$. Therefore, the signature x is eliminated like conventional generic signature schemes such as Schnorr [23] or Guillou-Quisquater [7]. In this case, the signature for m consists of (e, y).

6. Performance Evaluation

In this section, we evaluate the efficiency of our scheme by comparing existing fast signatures.

Table 1 gives the performance of various schemes, such as FFS, GPS, PS, OTM1 and OTM2, including our scheme. This table show that our signature scheme is quite efficient from both the computational cost and the transmitted data-size point of view.

Hereafter, we show the comparison between FFS (resp. GPS, resp. PS, resp. OTM1 and resp. OTM2) and our scheme.

FFS: FFS has a large size of public-key because at least $\Sigma_{i=1}^k |v_i| = k \times |n|$ bits are included in it. Similarly, our scheme also has a large size of public-key because of the same reason. As for the secret-key, FFS and our scheme have the same structure such as $s_i, 1 \leq i \leq k$. However, the size of secret-key in FFS is much larger than that in our scheme, because each s_i in FFS is large compared to our scheme. Consequently, FFS has the largest size for public-key of all the schemes in table 1.

GPS: Note that all the schemes in the table refer to under the one-key attack scenario [21]. Moreover, GPS has the provable security whose scheme is as secure as the discrete problem for modulo n. Those situations lead to the inefficiency in the size of secret-key: $|s| = 1024$. Consequently, GPS needs a large amount of work in the on-line phase: multiplication is 80×1024 on $\mathbb{Z}$. For more details, we refer to [17].

PS: PS has some drawbacks with respect to the structure of a secret-key. As you see in Section 2, the secret key in PS is created by $s = n - \phi(n)$. So the size of s is about half of $|s|$. Therefore it takes 80×512 in the on-line phase. On the other hand, since PS is intended to be used with a modulus product of two strong primes, $g = 2$ is a correct basis and does not have to be included in the public-key. Consequently, we can set the size of signature to 1024 bits for PS.

OTM1: One of the verifications in OTM is $x = g^{y-ze} \bmod n$. Let us now focus on the index of g in the equation. One can see the multiplication of two parameters z and e. On the other hand, the verification in our scheme is $x' = g^y \Pi_{i=1}^k v_i^{e_i} \bmod n$, so the multiplication in the index does not exist. the large-size index involved in the multiplication leads to the inefficiency from both the amount of work and data size point of view. Consequently, the amount of work for verification and the size of signature are both superior to those in OTM1.

OTM2: With respect to the modular reduction, we can use the efficient methods such as [1], [11] in implementation. Those methods are more advantageous than the multiplication computing in $\mathbb{Z}$ such as [8], because a single modulus can be used in the modular reduction and the other efficient reductions can be used. However, the addition is performed by using a more straightforward way and reduces the considerable amount of work compared to modular reduction. To sum up, the on-line additions in our scheme are faster than the on-line reduction in OTM2 from implementation point of view.

7. Conclusion

In this paper, we have proposed an efficient and fast signature scheme, which is derived from a three-pass identification scheme. As a result, the following remarkable advantages are given in our scheme. Firstly, the structure in the on-line phase is very simple: our scheme requires only hashing and addition. Consequently, minimal on-line computation is achieved. Secondly, the transmitted data size is small. We believe that our scheme would contribute to the cryptographic community.

We have also shown that our signature scheme is existentially unforgeable against any polynomial-time adversaries that can execute adaptive chosen message attack in the random oracle model. In this case, the underlying number theoretic problem is the integer factoring problem with an asymmetric basis g for an RSA modulus n.

References

[1] Paul Barrett. Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor. In A.M. Odlyzko, editor, *Proc. CRYPTO 86*, pages 311–323. Springer-Verlag, 1987. Lecture Notes in Computer Science No. 263.

[2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of 1st ACM Conference on Computer and Communications Security*, pages 62–73. Springer-Verlag, 1993.

[3] I. B. Damgård. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology - EuroCrypt '87*, pages 203–216, Berlin, 1987. Springer-Verlag. Lecture Notes in Computer Science Volume 304.

[4] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1:77–95, 1988.

[5] A. Fiat and A. Shamir. How to prove yourself: practical solutions of identification and signature problems. In *Crypto '86*, LNCS No. 263, pages 186–194. Springer-Verlag, 1987.

[6] M. Giraut. Self-certified public keys. In *Eurocrypt '91*, LNCS No. 547, pages 490–497. Springer-Verlag, 1992.

[7] L. C. Guillou and J. J. Quisquater. A "paradoxal" identity-based signature scheme resulting from zero-knowledge. In *Crypto '88*, LNCS No. 403, pages 216–231. Springer-Verlag, 1989.

[8] A. Karatsuba and Yu Ofman. Multiplication of multidigit numbers on automata. *Doklady Akademii Nauk SSSR*, 145(2):293–294, 1962.

[9] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1998. Third edition.

[10] D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, 1998. Second edition.

[11] P. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44:519–521, 1985.

[12] D. Naccache, D. MRaihi, D. Vaudenay, and D. Raphaeli. Can DSA be improved ? In *Eurocrypt '94*, LNCS No. 950, pages 77–85. Springer-Verlag, 1995.

[13] National Institute of Standards and Technology (NIST). Secure hash standard(SHS). In *Federal Information Processing Standards*, April 1995.

[14] T. Okamoto, H. Katsuno, and E. Okamoto. A fast signature scheme based on new on-line computation. In *Information Security Conference (ISC'03)*, LNCS No. 2851, pages 111–121. Springer-Verlag, 2003.

[15] T. Okamoto, M. Tada, and A. Miyaji. Proposal of efficient signature schemes based on factoring. In *Trans. IPSJ*, Vol.42 No. 8, pages 2123–2133, 2001 (in Japanese).

[16] T. Okamoto, M. Tada, and A. Miyaji. An improved fast signature scheme without on-line multiplication. In *Financial Cryptography '02*, LNCS No. 2357, pages 152–167. Springer-Verlag, 2002.

[17] D. Poincheval. The composite discrete logarithm and secure authentication. In *PKC '00*, LNCS No. 1751, pages 113–128. Springer-Verlag, 2000.

[18] D. Poincheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000.

[19] J. Pollard. Monte carlo methods for index computation (mod p). volume 32, pages 918–924. Mathematics of Computation, 1978.

[20] G. Poupard and J. Stern. Security analysis of a practical "on the fly" authentication and signature generation. In *Eurocrypt '98*, LNCS No. 1403, pages 422–436. Springer-Verlag, 1998.

[21] G. Poupard and J. Stern. On the fly signatures based on factoring. In *Proc. of the 6th CCS*, pages 48–57. ACM Press, 1999.

[22] Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.

[23] C. P. Schnorr. Efficient identification and signatures for smart cards. pages 239–252. Springer, 1990. Lecture Notes in Computer Science No. 435.