Witness Hiding Proofs and Applications

Lidong Chen

August 1994

Acknowledgements

I am deeply grateful to Professor Peter Landrock, my supervisor, for his constant encouragement and special understanding.

I am indebted to my office mate Doctor Torben Pryds Pedersen, who patiently and generously helped me in many ways.

I am indebted to Doctor Ivan Bjerre Damgård for his friendly assistance and bright explanations to many technical questions.

It has been an extremely pleasant experience doing research with the cryptology group at the Mathematics Institute of Aarhus University. The cosy academic atmosphere has been of vital importance to me during the accomplishment of my Ph.D program. All my colleagues are gratefully acknowledged.

I want to thank Doctor Mike Burmester for his constructive criticism and valuable advice during my visit to Royal Holloway, University of London.

I also want to express my thanks to the external referee, Doctor Stig Frode Mjølsnes, for reading this thesis very carefully and giving many comments and suggestions.

I would like to thank my husband, Jianshe, for his love and understanding during this venture.

Introduktion

Denne afhandling er baseret på mit Ph.D. studium i perioden 1991 - 1994, ved Aarhus Universitet, Danmark, med Peter Landrock som vejleder.

Hovedemnerne er dels studiet af en klasse af interaktive kendskabsbeviser, som har egenskaben at være såkaldte vidnesbeskyttende, dels deres anvendelser i realiserbare systemer. Nogle af de her repræsenterede resultater stammer fra samarbejde med Ivan Bjerre Damgård og Torben Pryds Pedersen.

Kryptologi er et forskningsområde, hvor mange andre felter bidrager, og det forudsættes, at læseren har godt kendskab til grundlæggende begreber indenfor kryptologi (specielt offentlig-nøgle systemer og digitale signaturer), talteori (så som kvadratiske rester og diskrete logaritmer), kompleksitetsteori (e.g. Turing maskiner), og informationsteori (entropi). Sidst i afhandlingen findes referenceliste som en hjælp til læsere, der ikke er bekendt med de ovenfornævnte begreber.

I et typisk kryptografisk scenarie må ejeren af en hemmelig nøgle gentagende gange demonstrere, at hun faktisk har adgang til nøglen, men uden at afsløre den. I et identifikationssystem bruges den hemmelige nøgle f.eks. til at identificere brugeren. I et robust kryptografisk system er det grundlæggende krav derfor, at den hemmelige nøgle forbliver "hemmelig", selv efter at have været brugt mange gange. Det var netop til det formål, at vidnebeskyttende interaktive protokoller blev introduceret: De sørger for at beskytte den hemmelige nøgle (vidnet) i en sådan grad, at hvis nøglen kompromiteres på en eller anden måde (som følge af den anvendte protokol), ville dette kunne opnås også selvom protokollen slet ikke var blev udført.

Kapitel 2 indeholder de grundlæggende definitioner og begreber omkring vidnebeskyttende protokoller. Den vidnebeskyttende egenskab er her defineret for generelle protokoller, som ikke nødvendigvis samtidig er kendsk-

absbeviser. Desuden introduceres en generel 3-skridts protokol i dette kapitel. Den er enten et kendskabsbevis eller bygget op som en protokol med det formål at give et bevis af en eller anden art. De fleste efterfølgende bevissystemer i denne afhandling er baseret på denne protokol. Ligeledes gives der en oversigt over digitale signatur systemer baseret på 3-skridts protokollen i dette kapitel.

Et centralt koncept i protokoller er det at kunne "aflede" (divertbility). En klasse af vidnebeskyttende protokoller for to parter siges at kunne afledes, hvis verifikatoren kan aflede udførelsen af protokollen til trediepart på en sådan måde, at verifikatoren kan udgive sig som beviseren. En nøjere analyse af begrænsninger for sådanne afledninger har både teoretisk og praktisk betydning. Kapitel 3 fokuserer på afledningsegenskaber ved vidnebeskyttende protokoller. Det bevises her, at en sådan protokol ikke kan afledes til mere end een uafhængig trediepart af en verfikator, som kan udføre polynomielle tidsalgoritmer, med mindre han kender vidnet. Dette har været et åbent spørgsmål i nogen tid.

En positiv anvendelse af afledning er såkaldte "blinde signaturer", på hvilket et anonymt akkrediteringssystem, der præsenteres i kapitel 4, er baseret. Ved brug af blinde signaturer kan akkreditiver udstedes på et pseudonym på en sådan måde, at den anonyme ejer kan overføre akkreditiverne fra et pseydonym til et andet. Endvidere kan akkreditiverne ikke forfalskes med mindre det diskrete logaritmeproblem kan løses.

Kapitel 5 vedrører praktisk anvendelige elektroniske valgsystemer. Vi foreslår her en formel definition af sådanne systemer og deres grundlæggende egenskaber, som muliggør en undersøgelse af elektroniske valgmodeler fra en ny vinkel. Endvidere præsenteres her to nye valgsystemer. Disse systemer anvender brug af såkaldte "bevismærker" (tokens), som udstedes til stemmeberettigede vælgere. Vi viser endvidere, at disse enten kan betragtes som et specielt eksempel på anonyme akkreditiver, eller drager en parallel til elektroniske mønt systemer.

Kapitel 6 starter med en formel definition af gruppesignaturer. I et sådant system kan ethvert medlem underskrive meddelelser anonymt på vegne af hele gruppen på en sådan måde, at identiteten er skjult i protokollen. Herpå præsenteres et nyt gruppesignatur system, baseret på vidnebeskyttende kendskabsbeviser. Disse nye gruppesignaturer tilbyder enten ubetinget eller beregningsmæssig anonymitet. Systemerne har endvidere den egenskab, at en myndighed eller en delmængde af gruppens medlemmer kan identificere un-

derskriveren forudsat kendskab til yderligere information stilles til rådighed, ved hjælp af hvilken den skjulte identitiet kan afsløres.

Med udgangspunkt i vores formelle definition af gruppesignaturer er det endvidere muligt at bevise nogle teoretiske resultater. Således vises det, at for at kunne underskrive flere meddelelser ubetinget anonymt, skal størrelsen af den hemmelige nøgle vokse. Samtidig vises det, at længden af den information nævnt ovenfor, der yderligere stilles rådighed ikke kan være mindre end tilstrækkeligt til at identificere underskriverne.

Contents

1 Introduction **11**

2 Basic Definitions and Concepts **15**
2.1 Interactive proofs of knowledge 15
2.2 Witness hiding 18
2.3 Three move witness hiding protocols 20
2.4 Zero-knowledge 24
2.5 Signature schemes 26

3 Parallel Divertibility **29**
3.1 Transferability and divertibility 29
3.1.1 General description 29
3.1.2 Related work 33
3.1.3 Definitions 33
3.1.4 Notation 36
3.2 Polynomial size E 39
3.2.1 2-extractable protocols 39
3.2.2 Z-extractable protocols ($l \geq 2$) 44
3.3 Exponential size E 45
3.4 Extensions 52
3.4.1 Transferability and divertibility with different inputs . 52
3.4.2 Divertibility of four move proof of membership 54
3.4.3 Blind signatures 57
3.5 Conclusions and open problems 57

4 Credentials with Pseudonyms **59**
4.1 Main idea and basic protocol 59
4.2 Validating pseudonyms 65
4.3 Issuing and transferring credentials 66
4.4 Unforgeability of credentials 68
4.5 Discussions 70

5 Practical Elections **71**
5.1 Introduction 71
5.2 Two typical practical election models 72
5.2.1 Hide voters' identities 72
5.2.2 Hide voters' votes 73
5.3 General election model 75
5.4 Voting under pseudonyms 79
5.5 Electronic cash based voting scheme 81
5.5.1 Voting coins 82
5.5.2 General coin-based voting 84
5.6 Discussion and further work 84

6 Group Signatures **87**
6.1 Introduction 87
6.2 Definitions 89
6.3 Proof of one out of n witnesses 93
6.4 Scheme with unconditional anonymity 95
6.4.1 Signing one message 95
6.4.2 Signing T messages 100
6.4.3 Identifying the signer 102
6.5 Lower bound on the secret keys 104
6.6 Length of the auxiliary information 106
6.7 Scheme with computational anonymity 108
6.8 Conclusion 112

Chapter 1

Introduction

This thesis was written as part of my Ph.D study at Aarhus University, Denmark with Peter Landrock as supervisor during the period of 1991-1994.

The main topics are on one hand the study of a class of interactive proofs of knowledge which has the the property of witness hiding and on the other hand applications in practical cryptology schemes. Some of the results presented are due to joint work with Ivan Bjerre Damgård and Torben Pryds Pedersen.

Cryptology is a subject where many other fields contribute, and the reader is assumed to have a good knowledge of basic concepts in cryptology (e.g. public key cryptology and digital signatures), basic number theory (e.g. quadratic residue and discrete logarithms), complexity theory (e.g. Turing machines) and information theory (e.g. entropy). A list of references will be given at the end of the thesis to guide the readers not quite familiar with the topics above to the right sources.

In a typical cryptographic scenario, a holder of a secret key must demonstrate repeatedly that she in fact knows the secret key, but without revealing it. For example, in an identification scheme, the secret key is used for identifying the user. In a robust cryptographic system, the basic requirement consequently is that the secret key remains "secret" even after being used many times. It is with this in mind witness hiding interactive protocols were introduced: They serve the purpose of protecting the secret keys to the extent that if the keys are compromised in some way this could have been achieved even without the protocol interaction.

Chapter 2 states the basic definitions and concepts concerning witness

hiding protocols. The witness hiding property is defined for the protocols which have a secret auxiliary input for the prover. A fundamental three move protocol is introduced. It is either a proof of knowledge or composed as a basic protocol to form a proof. Most proof systems in this thesis are based on it. A class of digital signature schemes based on the three move protocol is surveyed.

A central concept in protocols is that of divertibility. A class of two party witness hiding protocols is divertible in the sense that the verifier can divert the execution of the protocol to a third party in such a way that verifier plays the role of the prover. The investigation of the limitation of divertibility has both theoretical and practical significance. Chapter 3 focuses on the divertibility of witness hiding protocols. It proves that the witness hiding protocols cannot be diverted to more than one independent third party by any polynomial time verifier unless he knows the witness. This has been an open question for some time.

A positive application of divertibility is that of blind signatures, based on which, a credential system with pseudonyms is presented in Chapter 4. By using blind signatures, credentials may be issued on a pseudonym such that the anonymous owner can transfer the credentials from one pseudonym to another. At the same time, privacy of the user is protected unconditionally. Moreover, the credentials are unforgeable unless the discrete logarithm problem has an efficient solution.

Chapter 5 is concerned with practical elections. Formal definitions of practical elections and their basic properties are proposed. It enables to investigate election models at a new angle. Two new voting schemes are proposed. Such schemes employ the use of tokens issued to eligible voters. We show that these may either be considered a special example of credentials or be parallelized with electronic cash schemes.

Chapter 6 starts with a formal definition of group signatures. In group signature schemes, any member can sign messages anonymously on behalf of the group in such a way that the identity of the signer is hidden in the protocol. We then propose new group signature schemes based on witness hiding proofs of knowledge. The new schemes can provide either unconditional or computational anonymity. Moreover, the schemes include a general way to identify the signer in which an authority or a subset of group members can reveal the hidden identity of the signer by means of some auxiliary information.

Finally, some theoretical results are proved based on the formal definition stated earlier. It is shown that in order to sign more messages with unconditional anonymity, the size of secret keys must grow. Also the auxiliary information must contain enough to identify the signer.

Chapter 2

Basic Definitions and Concepts

In this chapter, we introduce the basic definitions and concepts behind witness hiding protocols.

2.1 Interactive proofs of knowledge

An interactive protocol between two parties is a sequence of data transmissions according to specific rules. A formal description of this is done by means of Turing machines.

Consider a pair of interactive Turing machines P and V. Each of them has its own work tape, WP and WV, a random tape, RP and RV, while they share a common read-only input tape I and a write-only output tape J. They communicate by two communication tapes: The write-only tape for P is read-only tape for V and the write-only tape for V is read-only tape for P. A complete description of a set of interactive rules for them is called a protocol and denoted as (P, V). We define a random variable $View^{(V)}_{P(x,w)}(x, y)$ to be V's view during the execution of the protocol with P on common input x, auxiliary input y for V, which consists of V's random coins and the messages received from P during executing the protocols. Similarly, we can define P's view $View^{(P)}_{V(x,w)}(x, y)$. The message written on the output tape by V (P) is called the output of V (P). We denote this as $V_{P(x,w)}(x)$ $(P_{V(x)}(x, w))$ or simply $V_P(x)$ $(P_V(x))$.

Interactive proofs were originally developed for membership in a language by Goldwasser, Micali and Rackoff (see [GMR89]). In such a proof one

party, $P(rover)$, is engaged in an interactive protocol with another party, $V(erifier)$. The task of the interaction is to convince the verifier that the input belongs to some language. In this case, it is a assumed that P has unlimited computational power.

Later, interactive proofs of knowledge were introduced(see [FFS88]). Suppose that $\mathcal{P}(x, w)$ is a polynomial time predicate[1]. For any x if w satisfies $\mathcal{P}(x, w) = 1$, then w is called a witness of x about $\mathcal{P}(x, w)$. The set of witnesses for x is defined as $w(x)$. The proof is for a polynomial time P to prove to V that he knows a witness w for input x.

Basic ingredients of any such protocols are *challenges* sent to the prover from the verifier and *responses* calculated by the prover, which are returned to the verifier for verification. In order to convince the verifier, the prover must be able to give correct responses to most challenges. In case of proof of knowledge, the prover's ability to reply to the challenges is due to possession of witnesses.

The formal definition of an interactive proof of knowledge was given by Feige, Fiat and Shamir (see [FFS88]) and Tompa and Woll (see [TW87]).

In this thesis, all proof systems discussed are proofs of knowledge, unless otherwise specified. We will point out when the discussions are suitable for proofs of language (membership) (see [GMR89]) as well or when the discrepancy between proofs of knowledge and proofs of language is important. For proofs of knowledge, the probabilistic Turing machines P and V as described above are supposed to access polynomial limited computational power only. Moreover, P and V each have access to a private knowledge tape KP and KV.

If k is the length of input, we use $\nu(k)$ to denote any function vanishing faster than the inverse of any polynomial in k. More formally,

$$\forall c \in N, \quad \exists k_0 \in N \ : \ \forall k > k_0, \quad 0 \leq \nu(k) < \frac{1}{k^c}$$

We characterize negligible probability as any probability function behaving as $\nu(k)$, and overwhelming probability to be a probability function behaving as $1 - \nu(k)$.

By nonnegligible probability, we mean that there exists a polynomial $p(k)$ such that when k is large enough the probability is larger than the inverse of

[1] A polynomial-time predicate $\mathcal{P}(x, w)$ is a predicate in which $|w|$ is polynomially related to $|x|$, and the truth value of the predicate can be checked in polynomial time.

$p(k)$. More formally, if the probability behaves as $\eta(k)$,

$$\exists c \in N : \ \exists k_0 \in N, \ \ \forall k > k_0, \ \eta(k) > \frac{1}{k^c}$$

Intuitively, an interactive proof of knowledge for predicate $\mathcal{P}(x, w)$ should satisfy the following, if the probability is a function of $|x|$.

- If P knows a witness w of x, then she should be able to convince V with overwhelming probability.
- If P does not know any w such that $\mathcal{P}(x, w) = 1$ then V can only be convinced with negligible probability.

As P is a Turing machine, what does it mean that P knows w? A possible assumption is that P has w in her knowledge tape. But it is too restrictive. P may know w in some strange way, for example, by guessing, computing from her knowledge tape. An informal definition of this concept was given in [GMR89].

> *P knows w if there is some polynomial time Turing machine M with complete control over P which prints w as a result of its interaction with P.*

Remark Here by "M with complete control over P" we mean that M is given the power to reset and rerun P polynomially many times without inspecting or modifying its tapes. Especially, when the protocol mainly consists of challenges from V and responses from P, M can send polynomial number of challenges to P in order to get responses, as V, did (probably only once) in one execution of the protocol.

We will follow the definition of Feige, Fiat and Shamir (see [FFS88]).

Definition 1 (Interactive proof of knowledge) A protocol (P, V) for polynomial time Turing machines P and V is called an interactive proof system for the polynomial-time predicate $\mathcal{P}(x, w)$ if it has the following properties.

1. *Completeness:* for all input x, if there exists w in P's knowledge tape such that $\mathcal{P}(x, w) = 1$, and both P and V follow the protocol, then V will output accept with overwhelming probability, *i.e.*

$$\forall a, \exists c, \forall [x] > c,$$
$$Prob[V_P(x) = accept] > 1 - \tfrac{1}{|x|^a}$$

2. *Soundness:* there exists a polynomial-time probabilistic Turing machine M with complete control on P such that for all P, any initial content of P's knowledge tape KP and random tape RP, and any efficient large $|x|$, if V can output accept with nonnegligible probability, then at the end of execution of $M(P, KP, RP)$, M can output a w' such that $\mathcal{P}(x, w') = 1$ with overwhelming probability, *i.e.*

$$\forall a, \exists M : \forall b, \forall P, \exists c, \forall [x] > c, \forall RP, \forall KP$$
$$Prob[V_P(x) = accept] > \frac{1}{|x|^a} \Rightarrow$$
$$Prob[output\ of\ M(P, RP, KP)\ on\ x\ satisfies \mathcal{P}] > 1 - \frac{1}{|x|^b}$$

In this thesis, a class of protocols will be considered which have a secret auxiliary input w for the prover and satisfy the completeness. They are defined as follows.

Definition 2 (Semi-proof) The protocol (P, V) for polynomial time bring machines P and V is called a semi-proof of knowledge for $\mathcal{P}(x, w)$, if it satisfies completeness.

2.2 Witness hiding

The purpose of proofs of knowledge is to convince the verifier that the prover knows some witness w for input x. If the prover must reply to some challenges from the verifier in the execution of the protocol, then a potential weakness of this procedure is that the verifier may use the prover as a so-called oracle. By choosing the challenges wisely, he can gain information about the witness which could not have been obtained just from the common input. An extreme example is to design the protocol simply so that P sends witness w to V for challenge *witness*? for common input x. V outputs *accept* if and only if $\mathcal{P}(x, w) = 1$. It is a proof of knowledge for the predicate $\mathcal{P}(x, w)$. But the witness is revealed. As a subroutine of most cryptographic schemes, the protocols must be secrecy protecting. The typical examples are identification schemes and signature schemes. Witness hiding is a formulation of this requirement.

Informally, a protocol (P, V) is witness hiding, if participating in the protocol does not help V to compute new witnesses to the input which he did not know at the begining of the protocol.

This concept was first formally defined by Feige and Shamir (see [FS90]) for proofs of knowledge. In this thesis, the witness hiding property will be defined for semi-proof of knowledge.

We suppose that $\mathcal{G}$ is a generator for the instances of the predicate $\mathcal{P}(x, w)$ which on input $1^{|x|}$ produces (x, w) such that $\mathcal{P}(x, w) = 1$.

Definition 3 (Witness hiding) Let (P, V) be a semi-proof of knowledge for a predicate $\mathcal{P}(x, w)$ and $\mathcal{G}$ be a generator. (P, V) is witness hiding on predicate $\mathcal{P}(x, w)$ and generator $\mathcal{G}$, if there exists a polynomial time limited Turing machine M such that for any polynomial time V, with complete control on V, M can extract a witness with the probability almost the same as the probability for V to output a witness, more precisely,

$$\begin{aligned}
&\exists M : \forall V, \forall a, \exists c, \forall [x] > c, \\
&Prob[V_{P(x,w)}(x) \in w(x)] < \\
&Prob[output\ of\ M(x, V, \mathcal{G}) \in w(x)] + \tfrac{1}{|x|^a}.
\end{aligned}$$

As we have mentioned, it is important to prove the witness hiding property for a protocol. But in most cases, in order to do that, another relevant property, witness indistinguishable, is first established for the protocol.

Informally, a protocol is witness indistinguishable if the verifier cannot tell which witness the prover is using (even if the verifier knows all witnesses to the statement being proved). When $w(x)$ only has one element, then witness indistinguishable property is trivial. It has been proved that if the protocol (P, V) is witness indistinguishable and $w(x)$ contains at least two independent witnesses, then the protocol must be witness hiding. The formal theorem and proof can be found in [FS90].

Definition 4 (Witness indistinguishable) Let (P, V) be a protocol for polynomial time predicate $\mathcal{P}(x, w)$. (P, V) is witness indistinguishable with respect to $\mathcal{P}(x, w)$ if for any V, any large enough $|x|$, any $w_1, w_2 \in w(x)$ and for any auxiliary input y for V, the ensembles, $View^{(V)}_{P(x,w_1)}(x, y)$ and $View^{(V)}_{P(x,w_2)}(x, y)$ generated as V's view of the protocol, are indistinguishable, more precisely,

$$\begin{aligned}
&\forall V, \exists c : \forall [x] > c, \forall KV, \forall w_1, w_2 \in w(x), \\
&View^{(V)}_{P(x,w_1)}(x, y) = View^{(V)}_{P(x,w_2)}(x, y).
\end{aligned}$$

Statistically (computationally) witness indistinguishable can be defined for the case that $View^{(V)}_{P(x,w_1)}(x,y)$ and $View^{(V)}_{P(x,w_2)}(x,y)$ are statistically (computationally) indistinguishable (see [GMR89]).

2.3 Three move witness hiding protocols

In this section, we consider a class of protocols which consists of three message transmissions. The common input of the protocol is denoted by x and E. The prover is convincing the verifier that he knows a witness w such that $\mathcal{P}(x,w) = 1$, where $\mathcal{P}(x,w)$ is a polynomial time predicate.

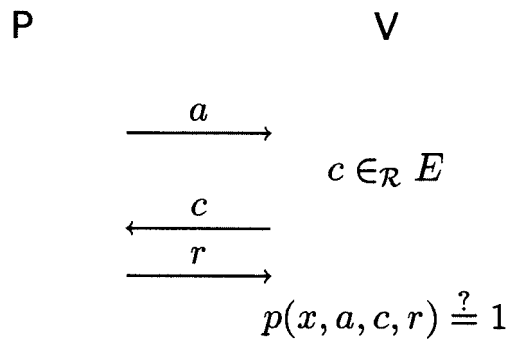

Figure 2.1: Basic three move protocol

As shown in Figure 2.1, the first message a is from the prover to the verifier, which, generally, is a function value of the prover's random coin which sometimes is called commitment. Then the verifier continues with a challenge c. Finally the prover sends the response r to the verifier. The verifier uses a polynomial time predicate $p(x,a,c,r)$ to decide whether to accept or reject.

This kind of protocols are basic protocols in most of the protocols appearing in this thesis. Also they are used as instances to demonstrate the concepts introduced in this chapter.

In our discussions, the size of the query set E is important. If k is the length of input of the basic protocol, we distinguish polynomial size E and exponential size E. By polynomial size E, we mean that there exists an integer c such that $|E| \leq k^c$. By exponential size E, we mean that $|E|$ is

larger than any polynomia1 of k.

We will call the three move protocol in Figure 2.1 as the basic protocol. If the basic protocol is a semi-proof of knowledge, and P knows the witness, then P must be able to reply to almost all the challenges in E correctly. Furthermore, if it is sound, then P shouldn't be able to reply to too many challenges in E without knowing witness. We are interested in the following property.

Definition 5 (Extractable) The basic protocol is *extractable*, if there exists an integer $l, 0 < l \leq |E|$, such that for any a, for any subset E_a of E satisfying $|E_a| \geq l$, there exists a polynomial time machine M with input

$$\{(c, r) \mid c \in E_a \;\; p(x, a, c, r) = 1\}$$

can output $w', \mathcal{P}(x, w') = 1$ with overwhelming probability.

The literature contains many examples of such three round protocols (e.g. in proofs for graph isomorphism, group membership, equality of discrete logarithms, quadratic residuosity).

Example Figure 2.2 shows a semi-proof of knowledge of square roots ([GMR89]) modulo $n = pq$. Here $E = \{0, 1\}$.

It is easy to establish completeness of the protocol. So the protocol is a semi-proof of knowledge of square roots modulo n.

In order to show that it is witness indistinguishable, assume that w and w' are both square roots of x modulo n. Denote $\delta = \frac{w}{w'}$. Then $\delta^2 = 1 \; mod \; n$. We will prove that with w and w', P will produce the same distribution. Let $View^{(V)}_{P(x,w)}(x, y) = (a, c, r)$, where $a = s^2 \; mod \; n$ and $r = sw^c \; mod \; n$. Also we notice that $a = s^2 = (s\delta^{-c})^2$ and $r = (s\delta^{-c})w'^2$. Since s's are chosen randomly in the protocol, P produces $View^{(V)}_{P(x,w')}(x, y) = (a, c, r)$ with the same probability.

It is extractable with $l = 2$. In fact, for any a, if $|E_a| \geq 2$, then $E_a = E$. From

$$\{(0, r), (1, r') \mid r^2 = a, r'^2 = ax\}$$

a square root of x, $w = \frac{r'}{r}$ can be computed.

However we observe that it does not satisfy soundness. P can succeed with probability $\frac{1}{2}$ without knowing the square root of x. In fact, P can

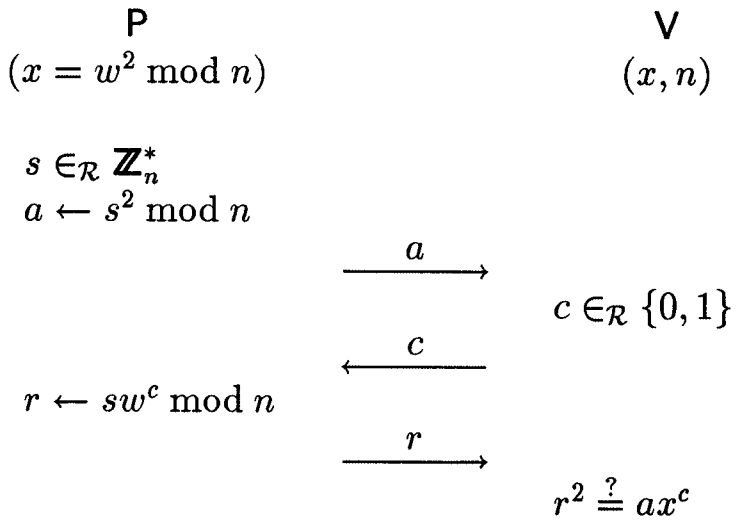

Figure 2.2: Semi-proof of a square root of x modulo n

choose $c' \in_R \{0,1\}$ and computes $a = s^2 x^{-c'}$. If the random challenge of V is c, $c = c'$, then P replying $r = s$ will get V outputing *accept*. So the probability of P succeeding is $\frac{1}{2}$.

It is common for many known three move witness hiding protocols that soundness is lacking. In order to achieve soundness, two basic comnositions are used:

1. **Sequential** The set E is in polynomial size. Execute the basic protocol t times sequentially. The verifier accepts the proof iff all t time executions of the basic protocol are accepted.

2. **Parallel** There are two types of parallel version of three move protocols.

   (a) The set E is of polynomial size. Execute the basic protocol parallelly with $a = (a_1, a_2, \ldots, a_t)$, $c = (c_1, c_2, \ldots, c_t)$, $c_i \in_R E$, $i = 1, 2, \ldots, t$, and $r = (r_1, r_2, \ldots, r_t)$. The verifier accepts the proof iff $p(x, a_i, c_i, r_i) = 1$, $i = 1, 2, \ldots, t$.

   (b) E is of exponential size, and the probability of successfully cheating is smaller than any polynomial inverse in a three move protocol.

The main difference between the two types of parallel versions is the communication complexity. The communication complexity of the first type is the same as the sequential one.

As subroutines of cryptographic schemes, the basic three move protocols are used in two versions: sequential and parallel, in order to construct a proof system. There are many concrete examples in literature (see [FFS88]) in which the composition protocol has been proved to be a proof.

Generally, it is easy to prove completeness for a basic three move protocol and obtain completeness of the composition protocol directly. We characterize several situations in which soundness can be proved. It is not hard to see that extractable is a key point to prove soundness.

Theorem 6 *If the basic three move protocol with polynomial size challenge set E is a semi-proof of knowledge, and it is extractable, then there exists $t = k^d$ for some positive integer d, such that the sequential version is a proof of knowledge for $x's$ witness w about the predicate $\mathcal{P}(x, w)$.*

Proof Suppose $|E| = k^c$ for some $c > 0$. Iterate the basic protocol t times with $t = k^{c+1}$.

In order to prove soundness, we show that whenever V accepts a prover P with nonnegligible probability, there exists a polynomial time extractor M such that M can print a witness for x with overwhelming probability.

Let T be the truncated execution tree of (P, V) for input x. V may ask $|E|$ questions in each stage. Each son of a vertex corresponds to one challenge to which P can reply successfully. A vertex is called heavy if it has $|E|$ sons. Since it is extractable, M can extract a witness of x from any of heavy vertices. The following is a proof of the fact that M can find a heavy vertex with overwhelming probability.

First, we show that at least half of the vertices in at least one of the levels in T must be heavy. Let α_i be the ratio between the number of vertices at level $i + 1$ and the number of vertices at level i in T. If $\alpha_i \leq (1 - \frac{1}{2|E|})|E|$, then the total number of leaves in T is bounded by $(1 - \frac{1}{2|E|})^t |E|^t$ which is a negligible fraction of the $|E|^t$ possible leaves. Since it is supposed that it is nonnegligible, there is at least one i such that $\alpha_i > (1 - \frac{1}{2|E|})|E|$ and thus at least half the vertices at this level must contain $|E|$ sons.

Then to find a heavy vertex T, M chooses polynomially many random vertices at each level, and determine their degrees by repeated resets and

execution of P. Since a nonnegligible fraction of leaves is assumed to survive the truncation, this exploration of T can be carried out in polynomial time. □

Remark In the proof of the theorem, t is not optimal. If the basic protocol is extractable with $\frac{l}{|E|} \leq \delta < 1$, where δ is a constant, then t can be any polynomial in k.

We will not distinguish between the first and second types of parallel versions in the following theorem.

Theorem 7 *If E is exponential size, and the basic protocol is a semi-proof of knowledge and it is extractable with $l \leq k^c$ for some positive , constant c, then the basic protocol is a proof of kiowledge of $x's$ witness w about predicate $\mathcal{P}(x, w)$.*

Proof sketch: If a prover P can succeed with nonnegligible probability, then for any knowledge tape KP and and random tape RP, a polynomial time extractor M can get l pairs of challenges and correct replies by reseting P polynomial number of times so that a witness can be extracted with overwhelming probability.

2.4 Zero-knowledge

For interactive proofs of knowledge, the most elegant concept is zero-knowledge, which captures the fact that an interaction proves the knowledge of a witness for the input but does not give any extra advantage to the verifier (see [GMR89]). In other words, what the verifier can produce after interacting with prover, he must be able to produce by himself beforehand. Regarding the formal definition of zero-knowledge, we will follow the definitions both given by Goldwasser, Micali and Rackoff (see [GMR89]) and by Feige, Fiat and Shamir (see [FFS88]). In [GMR89] it is defined for a general protocol relevant to membership of languages, and P has unlimited computational power. The definition of Feige, Fiat and Shamir is for proofs of knowledge. The following definition is for a general protocol but restricting the prover to have limited computational power only since this thesis considers this class

of protocols.

Definition 8 (Zero-knowledge) A protocol (P, V) is zero-knowledge if there exists a polynomial time Turing machine M, for all V and KV, and for all long enough input x, V's view $View_P^{(V)}(x)$ of the communication in (P, V) can be recreated by M as $View_M^{(V)}(x)$ such that $View_P^{(V)}(x)$ and $View_M^{(V)}(x)$ have the same distribution, more precisely,

$$\exists M : \forall V, \forall KV, \exists c, \forall x, |x| > c,$$
$$View_P^{(V)}(x) = View_M^{(V)}(x).$$

As an example, we prove that the basic protocol about square roots shown in Figure 2.2 is zero-knowledge. For any V, M works as follows:

1. M chooses $c' \in \{0, 1\}$ and $s \in \mathbb{Z}_n$ at random, computes $a = s^2 x^{-c'}$ and sends a to V.

2. M gets c from V,

   (a) if $c = c'$, sends $Y = s$ to V;

   (b) if $c \neq c'$, resets V, until $c = c'$.

   It is clear that M can produce $View_M^{(V)}(x) = (a, c, r)$ with the same distribution as $View_P^{(V)}(x)$ in expecting to reset V two times.

It is statistic (computational) zero-knowledge, if $View_P^{(V)}(x)$ and $View_M^{(V)}(x)$ are statistically (computationally) indistinguishable (see [GMR89]).

Zero-knowledge guarantees that no information whatsoever leaks during the execution of a protocol. But witness hiding only guarantees that the prover's witness does not leak, and says nothing about other information. It is clear that if a protocol (P, V) is zero-knowledge then it is witness indistinguishable (see [FS90]). But the concepts of witness hiding and witness indistinguishable have stronger practical background. The main reason is that the witness indistinguishable property is preserved under general compositions of protocols, but the zero-knowledge property is not, even though it can be preserved under sequential composition.

Especially, a digital signature scheme cannot be zero-knowledge, since a digital signature cannot be simulated by any polynomial time Turing machine, if it is unforgeable. But it should be witness hiding.

2.5 Signature schemes

In this thesis, we consider a class of digital signatures based on three move witness hiding protocols. The description of the basic construction is as follows.

Suppose that the three move protocol is a proof of knowledge. The witness is the prover's secret key. The basic idea is to use a hash value to substitute for the random challenge. The signature on message m is

$$\sigma(m) = (a, r)$$

satisfying $p(x, a, c, r) = 1$, where $c = \mathcal{H}(m, a)$ and $\mathcal{H}$ is a hash function behaving as a "random oracle". Regarding the security of signature schemes, this thesis follows the definition of Goldwasser, Micali, Rivest (see [GMR88]). It can be proved that if the basic protocol is a witness hiding proof of knowledge and the hash function behaves like a random oracle, then the signature is secure in the sense that it is protected against existential forgery under chosen message attack (see [GMR88]).

The first signature scheme of this style is proposed by Fiat and Shamir (see [FS87]). As an example, we present a signature scheme based on the proof of knowledge about square root modulo a composition.

Example

Considering the first type of parallel version of the basic protocol in Figure 2.2. We denote $\underline{a} = (a_1, a_2, \ldots, a_t)$, and $\underline{r} = (r_1, r_2, \ldots, r_t)$. The signature of message m is

$$\sigma(m) = (\underline{a}, \underline{r})$$

where $\underline{c} = (c_1, c_2, \ldots, c_l) = \mathcal{H}(m, \underline{a})$, satisfying

$$r_i^2 = a_i x^{c_i} \ mod \ n \quad i = 1, 2, \ldots, t,$$

for a hash function $\mathcal{H}$.

Remark There is another concept, *secure*, defined in [FFS88]. A protocol (P, V) is secure if after V participating the protocol polynomial number of times, the probability of V succeeding in playing the role of P to execute the protocol is still negligible. It can be proved that if (P, V) is a proof of

knowledge, then witness hiding and secure are equivalent properties for the protocol.

Remark Ohta and Okamoto had proposed a security measure for protocols in [OO90] called the *security level* ρ. In fact, the security level is a bound for the probability of success of a cheating prover. The relationship between security levels and other concepts, like witness hiding and secure, has been discussed in [CheDa93]. By introducing the property of extractable, it can be proved for some protocols that if the basic protocol is extractable with an integer l then the parallel version of the basic protocol has security level $\rho = \frac{1}{|E|}$.

Chapter 3

Parallel Divertibility

This chapter investigates the limitation of transferring and diverting certain interactive proofs. It is based on [CheDaPe94].

3.1 Transferability and divertibility

3.1.1 General description

By the definition of witness hiding proofs of knowledge, the prover P does not reveal any information which the verifier V can use to execute the protocol as P with V' offline. But the verifier can transfer or divert the protocol online to a third party. The following example demonstrates how this can be done.

Example

We consider the semi-proof of square root modulo a composite shown in Figure 2.2 in the previous chapter. The verifer V can divert the protocol as shown in Figure 3.1.

It is easy to see from the example that V can convince V' even though she does not know any square root of x. In this case, V uses P as an oracle. We observe that neither P nor V' can know what V has done.

As proofs of knowledge, the sequential and the first type of parallel version of the example can be transfered and diverted in the same way, *i.e.* V can convince V' that he knows a square root of x even though he does not know

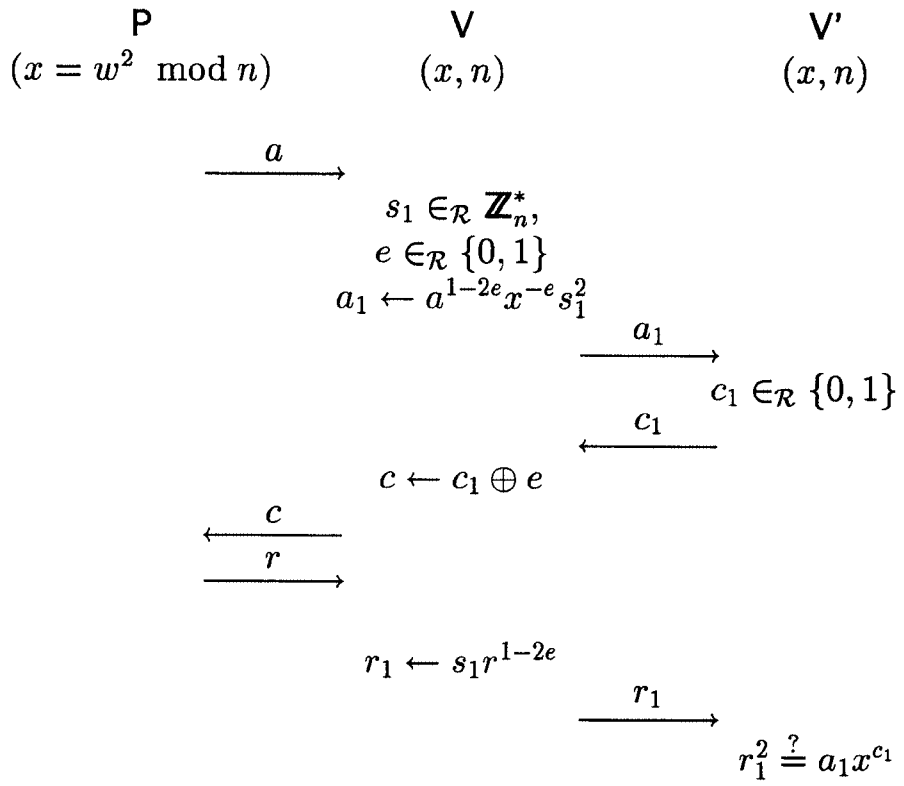

Figure 3.1: Diverting the semi-proof of a square root.

it at all. Due to this property, some cryptographic systems which uses the protocol as a subroutine can be abused.

If an identification protocol is based on this proof of knowledge, and it can be transfered as described above, then the so-called Mafia fraud is a threat for the system (see [DGB88]). Imagine V as a Mafia-owned shop. If user P usually buys her food from V, she proves her identity to V. At the same time, V transfers this proof to a jewelry shop V' and takes a piece of very expensive jewelry. Then P has to pay this without knowing who has got it.

The first important positive application of transferable and divertable protocols is to prevent subliminal channels (see [DGB88]). The concept of subliminal channel was first introduced by Simmons (see [Sim84]). The basic idea can be explained by an example: Two prisoners, P and V, are allowed to send messages in full view of a warden W. Consider an identification

scheme (P, V) based on the proof of square root. If P sends the message "I am Peggy" to V by executing (P, V), then additional messages can be sent without perceiving of W in the following way: instead of choosing $\underline{a}$ and/or $\underline{c}$ randomly, P and/or V can choose them as an encryption of some messages. In order to prevent this, W can divert the protocol in the way described above so that P and V cannot communicate by the subliminal channel, since the messages are changed by W. But P can still prove "I am Peggy" to V.

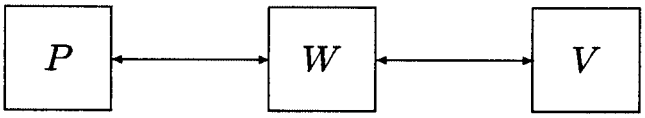

Figure 3.2: Three party protocol

For this historic reason, we will call the intermediary the warden. From now on, we use P, W and V to represent three parties in the transferable or divertible protocols as shown in Figure 3.2.

The blind signature was another practical application of transferable and divertible protocols suggested by Ohta and Okamoto (see [OO89]). We will not give a formal definition of blind signature but a general description.

> A blind signature scheme is a protocol between a signer and a receiver such that as a result of executing the protocol, the receiver can get a signature from the signer with the property that if the protocol is executed n times for n messages then the signer cannot recognize in which order he has signed the messages afterwards.

Blind signatures were first proposed for the purpose of electronic cash (see [Ch82]) to make the payment untraceable.

As an example, we show how to construct blind signatures by diverting the first type of parallel version of the protocol for square root in Figure 2.2. In order to sign a message m, W works as follows:

1. After getting

$$\underline{a} = (a_1, a_2, \ldots, a_t)$$

from P, W chooses

$$\underline{e} = (e_1, e_2, \ldots, e_t) \in_{\mathcal{R}} \{0,1\}^t \text{ and } \underline{s}_1 = (s'_1, s'_2, \ldots, s'_t) \in_{\mathcal{R}} (\mathbb{Z}_n^*)^t,$$

and computes

$$\underline{a}_1 = (a'_1, a'_2, \ldots, a'_t),$$

where

$$a'_j = (a_j^{1-2e_j}, x^{-e_j}(s'_j)^2,$$

$j = 1, 2, \ldots, t$.

2. W computes

$$\underline{c}_1 = \mathcal{H}(m, \underline{a}_1) = (c'_1, c'_2, \ldots, c'_t),$$

and sends

$$\underline{c} = \underline{c}_1 \oplus \underline{e}$$

to P, where $\oplus$ is bitwise plus.

3. When getting

$$\underline{r} = (r_1, r_2, \ldots, r_t),$$

from P, W computes

$$\underline{r}_1 = (r'_1, r'_2, \ldots, r'_t),$$

where

$$r'_j = r_j^{1-2e_j} s'_j$$

$j = 1, 2, \ldots, t$.

By executing the protocol with P, W can get a blind signature on a message m as

$$\sigma(m) = (\underline{a}_1, \underline{r}_1)$$

where $\underline{a}_1 = (a'_1, a'_2, \ldots, a'_t)$ and $\underline{r}_1 = (r'_1, r'_2, \ldots, r'_t)$, satisfying $(r'_j)^2 = a'_j x^{c'_j}$, $j = 1, 2, \ldots, t$, where $\underline{c}_1 = (c'_1, c'_2, \ldots, c'_t) = \mathcal{H}(m, \underline{a}_1)$, and $\mathcal{H}$ is a hash function.

While the ability to divert zero-knowledge proofs is very useful, it also has the effect that the prover can never be sure who will be convinced by his proof. It is therefore important to investigate to which extent it is possible to transfer and divert interactive proofs. The next section is a short review of the related work.

3.1.2 Related work

Divertibility was first introduced by Desmedt, Goutier, and Bengio in [DGB88]. In this paper, they pointed out how the Fiat-Shamir identification scheme (see [FS87]) can be abused by divertibility of the proof. Also they showed how to prevent subliminal channel.

[OO89] presented divertible proofs of knowledge for commutative, randomly self-reducible languages (see also [TW87]). These interactive proofs can also be regarded as proofs of membership and the divertible protocols work in that case as well, with the notable difference that for the proofs of knowledge both warden and verifier are convinced, whereas in the proof of membership the warden is only convinced under a computational assumption: if P has unlimited computational power and cooperate with V they can convince W a false statement. But even with unlimited computational power, P and W cannot cheat V.

However, this does not mean that the warden in general cannot be convinced unconditionally in divertible proofs of membership. In this chapter, a two round (four move) divertible, perfect zero-knowledge proof of membership is presented in which both warden and verifier are convinced unconditionally.

Further work on divertible proofs in [BD91] has resulted in divertible proofs for graph isomorphism and (given a probabilistic encryption homomorphism) for every language in NP (more precisely for SAT). Recently, [ISS93] constructed divertible proofs for graph non-isomorphism and a general protocol for every language in IP. However, these constructions seem to use a weaker definition of divertibility, and furthermore, the result for IP allows the verifier to send information to a necessarily unbounded prover.

All divertible proofs in [DGB88], [OO89] and [BD91] deal with (specific instances of) the three move protocol shown in Figure 2.1.

3.1.3 Definitions

Divertible proofs of knowledge were defined formally in [OO89]. As shown in Figure 3.2 the warden is an intermediary who, while interacting with P, convinces the verifier (if the verifier follows the protocol). In their definition, a divertible proof of knowledge must satisfy:

- If all three parties follow the protocol, the verifier will accept.
- The warden blinds the messages in such a way that neither the prover nor the verifier can tell the difference from an execution of the original proof system (even if they deviate from the protocol, see below).

In this section, first, transferability will be defined for a general protocol (P, V). The only requirement for transferability is that W can do what P is supposed to do in the protocol (P, V) by interacting with P in order to convince an honest verifier V. Then, parallel transferability will be introduced. By parallel transferability, we mean that W is not only trying to transfer the protocol to a single verifier, but to many verifiers, $V_1, V_2, \ldots, V_n$, given only one iteration with P, assuming that every V_i runs the protocol of V in (P, V) with W (see Figure 3.3). Finally, divertibility is defined as a special case of transferability by adding the requirement for the warden to blind the message.

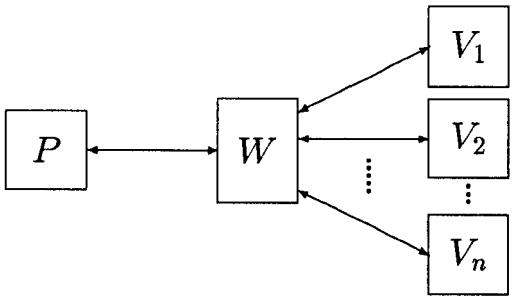

Figure 3.3: Parallel transferability.

If (P, V) is the protocol, we denote by $(P, W^{V_1,\ldots,V_n})$ the two party protocol between the prover and the warden, and by (W^P, V_i) that between the warden and the i'th verifier $(i = 1, 2, \ldots, n)$.

Remark There is a trivial way to transfer the protocol (P, V) to n verifiers $V_1, V_2, \ldots, V_n$, if the protocol is transferable. It can be done by V_1 transferring to V_2, V_2 transferring to V_3 and so forth, where V_i works as W not V. This kind of sequential transferability will not be considered here.

Definition 9 (n-transferable) Let (P, V) be a semi-proof for the predicate $\mathcal{P}(x, w)$. A warden, W is said to n-transfer the protocol (for $n \in \mathbb{N}$) with

probability π, if the following holds. Let $V_1, \ldots, V_n$ denote the n verifiers, let the common input to $P, W, V_1, \ldots, V_n$ be denoted by x and let P get w as auxiliary input, where $\mathcal{P}(x, w) = 1$. Whenever each V_i runs the protocol of the verifier in (P, V) with W independently it will accept with probability at least $\pi(x)$, where the probability is over the random coins of P, W and V_i.

The following notation is used:

- W is said to n-transfer a protocol if

$$\forall c, \exists k_0 \ : \ \forall x, |x| > k_0, \ \pi(x) \geq 1 - |x|^{-c}$$

- W is said to weakly n-transfer the protocol if

$$\forall c, \exists k_0 \ : \ \forall x, |x| > k_0, \ \pi(x) \geq |x|^{-c}$$

- The protocol (P, V) is called (weakly) n-transferable if there is a polynomial time warden, W, which (weakly) n-transfers it.

If the protocol (P, V) is a proof of knowledge, soundness for each V_i follows from soundness of the original (P, V)-proof system: V_i only accepts if W (using P as oracle) knows a witness.

According to the definition, every semi-proof is 1-transferable, because the warden can just forward P's messages to V and vice versa. But it is not necessarily divertible according to the definition of divertibility of [OO89]. For divertibility, it is required that the warden must blind the messages such that the random variables

$$(\mathit{View}_W^{(P)}(x, s), \mathit{View}_W^{(V)}(x, y))$$

and

$$(\mathit{View}_V^{(P)}(x, s), \mathit{View}_P^{(V)}(x, y))$$

are indistinguishable, *i.e.* neither P nor V can recognize whom she or he is talking with, W or V (P) even if they deviate from the protocol. But we suppose that the prover is able to send the correct answer to warden. It is not meaningful to expect the warden to divert a proof if he does not receive one.

Definition 10 (n-divertible) Let (P, V) be a proof of knowledge. The protocol is n-divertible if there is a polynomial time warden, W, such that

1. W n-transfers (P, V);

2. For any prover $\tilde{P}$ and any n verifiers $\tilde{V}_i$ $(i = 1, 2, \ldots, n)$ for which there is a $c > 0$ such that for $|x|$ sufficiently large $\tilde{P}$ convinces an honest verifier in (P, V) with probability at least $1 - |x|^{-c}$ the following holds:

$$(View_W^{(\tilde{P})}(x, s),\ View_W^{(\tilde{V}_1)}(x, s_1), \ldots,\ View_W^{(\tilde{V}_n)}(x, s_n))$$

and

$$(View_V^{(\tilde{P})}(x, s),\ View_P^{(\tilde{V}_1)}(x, s_1), \ldots,\ View_P^{(\tilde{V}_n)}(x, s_n))$$

and are indistinguishable for $|x|$ sufficiently large.

It is perfectly (statistically, computationally) n-divertible, if the two random variables above are perfectly (statistically, computationally) indistinguishable. (In case of computationally n-divertible, the cheating provers must be polynomially bounded). In this thesis, by "indistinguishable", we mean perfectly indistinguishable.

It is an immediate consequence of the definitions that if a proof cannot be transferred then it cannot be diverted.

The definition puts no restraints on the order of the messages which W sends to the n verifiers. For example, in one extreme, W first diverts the proof to V_1 and then, afterwards, to V_2 and so forth. In an other extreme W computes the messages to V_i depending on the messages from not only P, but the other verifiers as well.

3.1.4 Notation

The discussion in this chapter mainly deals with the protocols based on witness hiding basic three move protocol shown in Figure 2.1. Two situations are concerned:

1. The basic protocol is a semi-proof of knowledge and can be iterated to get a proof of knowledge and E has polynomial size.

2. The basic protocol is a proof of knowledge and E has exponential size.

In the three move protocol, a denotes the initial message from P, c the challenge from V, and r the reply from P.

Generally, the warden computes the messages to P and V as functions of the messages he received and his random bits ρ. The functions for computing the initial message, the challenge and the reply are denoted by f, g, and h respectively (see Figure 3.4).

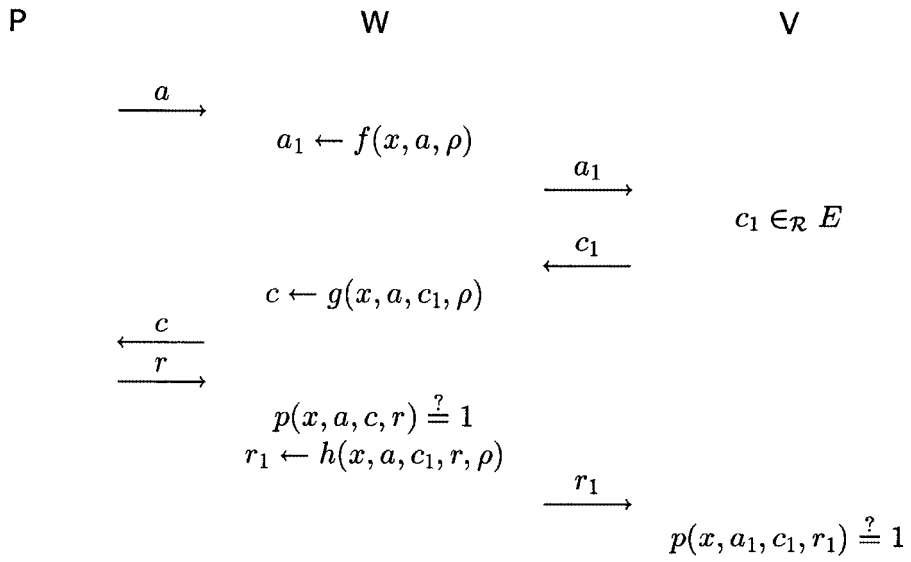

Figure 3.4: Transferring or diverting the basic protocol.

Intuitively, as mentioned before for the prover P, in order to be successful in transferring or diverting W must be able to answer many challenges, c_1's, from V. Given functions f, g, and h, for any c_1, whether c_1 can be answered depends on both a and r and random bits ρ. So the following set is important.

Definition 11 Given three functions f, g, h as in Figure 3.4. For any a, r and ρ, define a set $S_{\rho,a,r}$ as the set of c_1's which can be answered. Preciseiy, $c_1 \in S_{\rho,a,r}$ if only if

$$p(x, a, g(x, a, c_1, \rho), r) = 1$$

and

$$p(x, f(x, a, \rho), c_1, h(x, a, c_1, r, \rho)) = 1$$

When considering the possibility of transferring or diverting the basic protocol to two verifiers in parallel we use the notation shown in Figure 3.5.

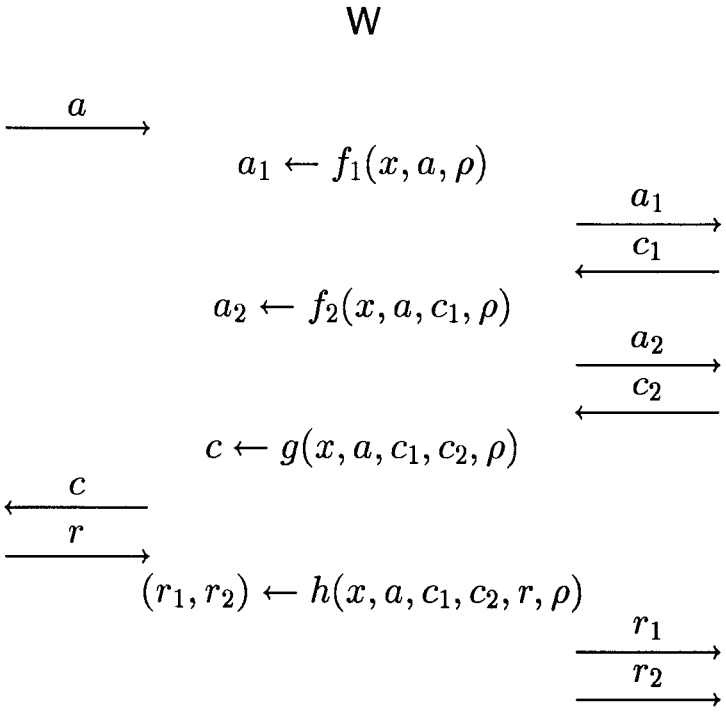

Figure 3.5: Notation for parallel divertibility.

The meaning of this should be self-explanatory except for the order of the messages. We allow the warden to compute the initial value a_2, to be sent to V_2 depending on the challenge c_1 chosen by V_1. This is necessary as it is unreasonable to require any synchronisation between the two independent verifiers V_1 and V_2 (in fact V_1 may not be aware that the proof is being transferred to V_2). We also require that W receives a challenge from both verifiers before computing the challenge, c, to P. This makes the warden most general as the function g can always ignore some of its inputs.

Alternatively, the warden could postpone computing a_2 until it has received r from the prover. However, then the warden would be able to execute the protocol as a prover after one execution of the protocol with P.

3.2 Polynomial size E

In this section, we will prove that the basic protocol with polynomial size challenge set E is not weakly 2-transferable. The proof depends on extractable property of the protocol.

In the previous chapter, the extractable property has been defined for three move protocols (see Definition 5). By the definition, if the basic protocol is extractable, then there exists an integer l such that for any initial message a, and any subset E_a of E, if $|E_a| \geq l$, a witness w' can be computed by a polynomial time Turing machine from the set

$$\{(c, r) \mid c \in E_a,\ p(x, a, c, r) = 1\}$$

In this chapter, the number l is so important that we prefer to denote the property l-extractable.

First, the protocol which is 2-extractable will be investigated. Then a general result for l-extractable protocols is given. Theoretically, there is no difference between these two situations. But most known protocols are 2-extractable and can be proved not 2-transferable. Furthermore, the general result for extractable protocols will reveal the interesting fact that in some cases the warden may be able to transfer the protocol to more than one verifier without knowing the witness.

3.2.1 2-extractable protocols

In this section, we will prove that if the basic protocol is 2-extractable, then no polynomial time warden can transfer it to two verifiers with probability larger than $\frac{1}{|E|}$ unless he knows a witness. It can be imagined intuitively that the warden can transfer the protocol to one person with probability 1, but for another, the probability of success cannot be larger than that he executes the protocol by himself.

Lemma 12 *Consider a 2-extractable basic protocol. For any three functions (f, g, h) used by the warden to transfer the basic protocol to a single verifier the following holds: if for some $d > 0$ and k suficiently large with probability at least k^{-d} there exist $c_1, c_1' \in S_{\rho,a,r}(c_1 \neq c_1')$ such that*

$$g(x, a, c_1, \rho) = g(x, a, c_1', \rho)$$

(the probability is over the choices of (a, r) by the prover and ρ), then for such k's there is an $e > 0$ and a probabilistic polynomial time verifier which can compute w' satisfying $\mathcal{P}(x, w') = 1$ with probability at least k^{-e} after one execution of the basic protocol.

Proof Assume the protocol is 2-extractable and let (f, g, h) be the three polynomial time functions used by the warden. We will construct a polynomial time verifier, M, which using (f, g, h) extracts P's witness after one execution of the protocol. M computes, given a from the prover, its challenge as follows:

1. Choose a random string, ρ, of the proper length.
2. Produce a table of pairs (c, c_1) for all $c_1 \in E$, where $c = g(x, a, c_1, \rho)$.
3. If there is a c such that (c, c_1) and (c, c_1') $(c_1 \neq c_1')$ occur in the table, then return a random such c, otherwise return a random $c \in E$.

Given r from the prover, M computes

$$r_1 = h(x, a, c_1, r, \rho) \quad \text{and} \quad r_1' = h(x, a, c_1', r, \rho).$$

Let $a_1 = f(x, a, \rho)$. If $p(x, a_1, c_1, r_1) = p(x, a_1, c_1', r_1') = 1$ then the machine, guaranteed by 2-extractable, can be used to extract the witness. It is easy to see that M runs in polynomial time and it succeeds if and only if it finds $c_1, c_1' \in S_{\rho,a,r}$ satisfying

$$g(x, a, c_1, \rho) = g(x, a, c_1', \rho)$$

which happens with probability at least a polynomial fraction. □

This lemma says that no warden can compute correct responses to two different challenges from one execution of the basic protocol with P.

Remark This proof only works when E is of polynomial size, since it is infeasible to search the table of pairs (c, c_1) for all $c_1 \in E$ in polynomial time when E is of exponential size.

For the proof of our main result, we also need the following simple lemma.

Lemma 13 *Let E be a finite set, $D \subseteq E \times E, |D| \geq |E| + 1$, and σ a function from D to E. If σ has the property that*

$$\forall (x_1, y_1), (x_2, y_2) \in D \ : \ x_1 \neq x_2 \Rightarrow \sigma(x_1, y_1) \neq \sigma(x_2, y_2),$$

then there exist $(x, y_1), (x, y_2) \in D$, such that $y_1 \neq y_2$ and $\sigma(x, y_1) = \sigma(x, y_2)$.

Proof If

$$\forall (x_1, y_1), (x_2, y_2) \in D \ : \ x_1 \neq x_2 \Rightarrow \sigma(x_1, y_1) \neq \sigma(x_2, y_2),$$

then if it is also true that for all $(x, y_1), (x, y_2) \in D, y_1 \neq y_2$,

$$\sigma(x_1, y_1) \neq \sigma(x_2, y_2),$$

σ is $1-1$ from D to E. Since $|D| > |E|$, and both of them are finite sets, it is impossible. So there exists (x, y_1) and (x, y_2) in D, $y_1 \neq y_2$ such that $\sigma(x, y_1) = \sigma(x, y_2)$. □

Lemma 14 *If the basic protocol is 2-extractable and witness hiding, and $|E| \leq k^d$ for some integer d, then no polynomial time warden can transfer it to two verifiers, V_1 and V_2 in parallel and answer more than $1/|E|$ of the possible pairs of challenges with non-negligible probability. The probability is over the choices of (a, r) and ρ.*

Proof The idea is that if a warden can answer more than $1/|E|$ of the possible pairs of challenges then, by Lemma 13, it can answer two different challenges from V_1 or V_2 using the same challenge to the prover. Since the protocol is 2-extratable, one of the witnesses can be extracted. This contradicts the property of witness hiding. The following makes this more precise.

Let (f_1, f_2, g, h) be the functions used by the warden (see Figure 3.5). A new warden, W' will be constructed which uses these functions to divert the protocol to a single verifier. Note however, that the warden a priori does not know whether it can answer two challenges from V_1 or V_2. To overcome this problem the new warden decides at random whether to divert it to V_1 or V_2. Let the functions used by W' be (f', g', h'). We have to specify these functions.

1. Choose $b \in \{1, 2\}$ at random (W' will use f_b). Also choose a random string $\rho \in \{0, 1\}^*$.

2. If $b = 1$ the three functions are computed as follows:

   - $f'(x, a, \rho) = f_1(x, a, \rho)$.
   - Given c' from the verifier $g'(x, a, c', \rho) = g(x, a, c', c_2, \rho)$ where $c_2 \in E$ is chosen at random.
   - Compute $(r_1, r_2) = h(x, a, c', c_2, r, \rho)$ and let $h'(x, a, c', \rho) = r_1$.

   If $b = 2$ the functions are

   - Compute $f'(x, a, \rho) = f_2(x, a, c_1, \rho)$, where $c_1 \in E$ is chosen at random.
   - Given c' from the verifier $g'(x, a, c', \rho) = g(x, a, c_1, c', \rho)$.
   - Compute $(r_1, r_2) = h(x, a, c'_2, c', r, \rho)$ and let $h'(x, a, c', \rho) = r_2$.

Clearly, each of these functions can be computed in polynomial time. Let *Succ* denote the event that W succeeds answering more than $|E| + 1$ of the possible pairs of challenges and let π denote the probability of this event (over the choices of (a, r) and ρ).

Now assume that *Succ* occurs and let D denote the set of challenges that W can answer. If there exists two pairs $(c_1, c_2), (c'_1, c'_2) \in D$ where $c_1 \neq c'_1$ then W can answer two different challenges from V_1 corresponding to the same a_1. If two such pairs do not exist then, by Lemma 13 there exist two different pairs $(c_1, c_2), (c_1, c'_2) \in D$. The challenges c_2 and c'_2 will correspond to the same a_2 (as $a_2 = f_2(x, a, c_1, \rho)$).

Thus if *Succ* occurs, W can answer two different challenges from either V_1 or V_2 using the same challenge to P. Let $A_e(a, r, \rho)$ denote the event that W can answer two different challenges from V_e, $e = 1, 2$, for (a, r, ρ) by using the same challenge to P.

If W' diverts the protocol to V_e then for $e' \neq e$, instead of getting a challenge from $V_{e'}$, he chooses $c_{e'} \in_{\mathcal{R}} E$. It is called a good choice if for $c_{e'}, W'$ can answer two challenges from V_e by the same challenge to P. Let $C_e(a, r, \rho)$ denote the event that W' gets a good choice for given (a, r, ρ) when diverting the protocol to V_e. (a, r, ρ) will be omitted from $A_e(a, r, \rho)$ and $C_e(a, r, \rho)$ in the following.

Assume p is the probability that W' can answer two different challenges from V using the same challenge to P. Then for any e,

$$\begin{aligned} p &\geq Prob[C_e \wedge A_e \wedge b = e \wedge Succ] \\ &\geq Prob[C_e | A_e \wedge b = e \wedge Succ] Prob[A_e \wedge b = e | Succ] Prob[Succ] \\ &\geq \frac{\pi}{2|E|} \end{aligned}$$

Thus if π is larger than the inverse of a polynomial then the probability that W' can answer two different challenges from V using the same challenge to P is nonnegligible. Since the basic protocol is 2-extractable, by Lemma 12, the witness can be extracted in polynomial time with nonnegligible probability, which gives a contradiction with the property of witness hiding. □

The following theorem extends this lemma to cope with the application of the basic protocol to proofs of knowledge.

Theorem 15 *Assume no polynomial time algorithm can find a witness, w, such that $\mathcal{P}(x, w) = 1$. Let (P, V) be a witness hiding proof of knowledge, in which the basic protocol is repeated t times where $1 < |E| < k^d$ for some d and $|E|^t$ grows faster than any polynomial in k. If the basic protocol in 2 extractable then (P, V) is not weakly 2-transferable.*

Proof Since the basic protocol is witness hiding, by Lemma 14, for any warden, the probability of success in transferring the basic protocol to two verifiers V_1 and V_2 is at most $1/|E|$. Thus the probability of success in t independent iterations is at most $(1/|E|)^t$ which by assumption is smaller than the inverse of any polynomial when k is large enough. □

Remark Theorem 15 includes zero-knowledge proofs as a special situation. In fact, most known basic three move protocols with polynomial size E are zero-knowledge. So the proof systems constructed by iterating the basic protocol t times are zero-knowledge proofs.

3.2.2 Z-extractable protocols ($l \geq 2$)

Most known three-move witness hiding protocols are 2-extractable. However, the general result regarding l-extractable has theoretical significance. We will see how much the proof depends on l and in some cases, the warden can transfer the protocol to more than one verifier.

Theorem 16 *Let $n \in \mathbb{Z}$ be a constant (independent of k), and let l be a polynomial in k. If $(l-1)^n|E| < |E|^n$, then no warden can answer more than $(l-1)^n|E|$ of the possible tuples of challenges with non-negligible probability in n-transferring a witness hiding l-extractable basic protocol in which $|E| < k^d$ for some integer d.*

Proof We will just point out how the proof of Lemma 14 must be changed.

As $|E|$ is polynomial size, the same proof as for Lemma 12 shows that no warden can answer more than l challenges from the verifier using the same challenge to the prover.

Let W be the warden who n-transfers the proof. Assume that the warden computes the initial message a_i to V_i (for $i > 1$) depending on the challenges from $V_1, \ldots, V_{i-1}$ (a natural extension of the case $n = 2$).

As in the proof of Lemma 14 it is possible to construct a new warden, W', which with probability at least the inverse of a polynomial can transfer the proof to a single verifier and answer more than l challenges using the same challenge to the prover.

To do this note that, if W can answer more than $(l-1)^n|E|$ tuples of challenges then for some i the warden can answer at least l tuples of the form

$$(c_1, \ldots, c_{i-1}, c_i^{(j)}, c_{i+1}^{(j)}, \ldots, c_n^{(j)})$$

for $j = 1, 2, \ldots, l$ and where

$$c_i^{(j)} \neq c_i^{(j')}) \quad \text{for} j \neq j',$$

by using the same challenge c to P. This extension of Lemma 13 follows from a simple counting argument (these l tuples will all give rise to the same initial message a_i to V_i).

Let π denote the probability of the event that W can answer more than $(l-1)^n|E|$ tuples of challenges.

W' now works very much along the lines of the proof of Lemma 14. W' first guesses i and then $c_1, \ldots, c_{i-1}$. From these it can compute the initial message to the verifier. Then given a challenge c' from the verifier it chooses $c_{i+1}, \ldots, c_n$ at random and computes a challenge to the prover. From the reply to this challenge W' can extract the reply to the verifier.

If W can answer more than $(l-1)^n|E|$ tuples of challenges, W' is lucky, if he chooses i and $c_1, \ldots, c_{i-1}, c_{i+1}^{(j)}, \ldots, c_n^{(j)})$, such that they can be used with l different c_i-values then W' can answer l different challenges corresponding to the same initial message by using the same challenge to P. The probability that W' is lucky is at least

$$\frac{\pi}{n}|E|^{1-n}$$

which is the inverse of a polynomial. □

This theorem shows that a warden can answer correctly at most a fraction $(l-1)^n|E|^{1-n}$ of the possible challenges. If for some t polynomial of k, $[(l-1)^{-n}|E|^{n-1}]^t$ grows faster than any polynomial of k, then the proof system constructed by iterating the basic protocol t times is not weakly n-transferable.

Example For $n = 2, l = 3$ and $|E| = 5$ this fraction is $4/5$. Hence, in this case the proof cannot be 2-transferred with very high probability.

Example For $n = 2, l = 3$ and $|E| = 4$ this fraction is 1 — hence the theorem does not exclude 2-transferability in this case. However, the protocol cannot be 3-transferred: the fraction is $2^3/4^2 = 1/2$.

3.3 Exponential size E

While the previous section assumed that $|E|$ is polynomial in k, this section considers the case where a single execution of the basic protocol constitutes a proof system. Hence, $|E|$ is larger than any polynomial for k sufficiently large. It will be shown that this protocol is not 2-divertible if it is witness hiding. In particular, this shows that the usual ways (e.g., as in [OO89]) of diverting instances of this protocol to a single verifier cannot be extended to parallel divertibility, if the warden still wants to be successful with overwhelming probability.

The proof in the previous section depends on E being of polynomial cardinality and we see no way to modify it to cope with larger E's. Therefore this section will use another technique, which does neither seem to work for transferability nor seem to extend to the case of weak divertibility (as will be pointed out). However, on the positive side the result in this section does not require the extractable property.

First, two lemmas are needed. Consider the situation where W tries to divert the proof to a single verifier, V. Lemma 17 below shows that W cannot use the same c in order to answer too many challenges from V. This lemma can be regarded as an extension of Lemma 12 to the case of large E.

Lemma I7 *For any warden given by (f, g, h) the following holds. If the protocol is a witness hiding proof of knowledge, then for all $d, e > 0$ the probability that there is an $c \in E$ such that*

$$|\{c_1 \in S_{\rho,a,r} \mid g(x, a, c_1, \rho) = c\}| > |E|/k^d$$

is at most k^{-e} for k suficiently large. This probability is over the choices of (a, r) and ρ.

Proof Let (f, g, h) be given and assume there are $d, e > 0$ such that for infinitely many values of k the cardinality of the above set is larger than $|E|/k^d$ with probability larger than k^{-e}. A verifier will be described which on input x of length such a k can find a witness after one execution of the protocol. As the protocol is a proof of knowledge it is sufficient to construct a verifier, M, which after a single execution can convince an honest verifier with probability at least $k^{-(e+2d)}$. M works as follows

1. Given a from the prover, compute $a_1 = f(x, a, \rho)$.
2. Choose $c_1 \in E$ at random and compute $c = g(x, a, c_1, \rho)$.
3. Get r from the prover and compute $r_1 = h(x, a, c_1, r, \rho)$.
4. If $p(x, a_1, c_1, r_1) = 0$ then stop.

Afterwards M, when acting as a prover, chooses a_1 in the first move, and given a challenge c_1' from the honest verifier returns

$$r_1' = h(x, a, c_1', r, \rho).$$

Clearly, M runs in polynomial time. In the following it will be shown that M convinces the verifier with probability at least $k^{-(e+2d)}$. Let for $c \in E$

$$E_c = \{c_1 \in S_{\rho,a,r} \mid g(x, a, c_1, \rho) = c\},$$

and let A denote the event that there is a c such that

$$|E_c| > |E|/k^d.$$

Now

$$\begin{aligned} Prob[V \text{ accepts}] &\geq Prob[V \text{ accepts}|A, c_1 \in E_c]Prob[c_1 \in E_c|A]Prob[A] \\ &\geq Prob[V \text{ accepts}|A, c_1 \in E_c]k^{-d}k^{-e} \\ &\geq Prob[c_1' \in E_c]k^{-(e+d)} \\ &\geq k^{-(e+2d)}. \end{aligned}$$

□

The next lemma shows that if W can divert the proof to two persons in parallel, then she can divert it to any polynomial number of verifiers. This lemma fails in the cases of transferability and weak divertibility.

Lemma 18 *If an interactive proof system is 2-divertible then it is also n-divertible for any n which is polynonaial in k.*

Proof Let $n = n(k)$ be polynomial in k. The method for W to divert the proof to two verifiers can be used in a tree-like way to divert it to n verifiers as shown in Figure 3.6. Each W_i works as W. In this case, the new warden is the combination of $W_1, W_2, \ldots, W_n$, who divert the protocol to n verifiers under the definition of parallel divertibility.

Let for a given $k, \pi_0(k)$ be the probability that W fails to divert the proof when talking with the prover, and let $\pi_i(k)$ be the probability that W_i fails. Then if the proof is 2-divertable and W_i works as W, then for any i, $\delta_i(k) = |\pi_0(k) - \pi_i(k)|$ is less than the inverse of any polynomial for k sufficiently large (otherwise the proof of one of the wardens can be distinguished from that of the prover).

For any $1 \leq j \leq n$, it fails to divert the proof to V_j, if and only if at least one of the wardens W_i who is on the pass to V_j in the tree structure of Figure 3.6 fails to divert. Thus for any j, the probability that the proof

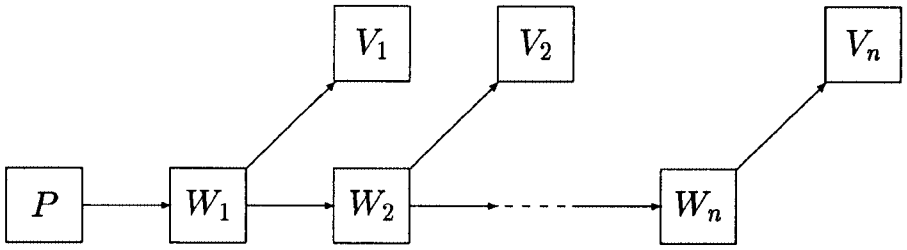

Figure 3.6: Diverting a proof to many verifiers

fails for V_j is at most $n(\pi_0(k) + \delta(k))$, where $\delta(k) = max\{\delta_i(k)\}$. For any $d > 0$, this probability can be made less than k^{-d} for k sufficiently large. For the combined new warden W, it must be proved that the views are indistinguishable. For simplicity, the proof for $n = 3$ is given. It can be extended to all $n = n(k)$, polynomial in k. In order to prove

$$H_0 = (\,View_W^{(\tilde{P})}(x, s),\, View_W^{(\tilde{V}_1)}(x, s_1),\, View_W^{(\tilde{V}_2)}(x, s_2),\, View_W^{(\tilde{V}_3)}(x, s_3))$$

and

$$H_3 = (\,View_V^{(\tilde{P})}(x, s),\, View_P^{(\tilde{V}_1)}(x, s_1),\, View_P^{(\tilde{V}_2)}(x, s_2),\, View_P^{(\tilde{V}_3)}(x, s_3))$$

are indistinguishable for any $\tilde{P}$ who can convince an honest verifier with overwhelming probability and any verifiers $\tilde{V}_i, i = 1, 2, 3$, a hybrid argument (see [GM84a]) is used.

Consider two hybrids,

$$H_1 = (\,View_W^{(\tilde{P})}(x, s),\, View_W^{(\tilde{V}_1)}(x, s_1),\, View_W^{(\tilde{V}_2)}(x, s_2),\, View_P^{(\tilde{V}_3)}(x, s_3))$$

and

$$H_2 = (\,View_W^{(\tilde{P})}(x, s),\, View_W^{(\tilde{V}_1)}(x, s_1),\, View_P^{(\tilde{V}_2)}(x, s_2),\, View_W^{(\tilde{V}_3)}(x, s_3)).$$

The argument is that if each pair of random variables (H_{i-1}, H_i), $(i = 1, 2, 3)$, is indistinguishable, then H_0 and H_3 are indistinguishable.

Before the proof, notice that any prover $\tilde{P}$, involved in the divertible proof, together with any $\tilde{V}_i$, can produce instances of $View_P^{\tilde{V}_i}(x, s_i)$, $i = 1, 2, 3$, with overwhelming probability, since the protocol is a proof of knowledge and

any prover $\tilde{P}$ who can convince an honest verifier with overwhelming probability, as required in the definition of divertibility, must know the witness (with very high probability).

First, the main idea of proving that H_0 and H_1 are indistinguishable is the following: if they are distinguishable, a new combined prover $\tilde{P}^*$ can be constructed for which

$$H_{W_3} = (\mathit{View}_{W_3}^{(\tilde{P}^*)}(x, s^*), \mathit{View}_{W_3}^{(\tilde{V}_3)}(x, s_3),$$

and

$$H_V = (\mathit{View}_{V}^{(\tilde{P}^*)}(x, s^*), \mathit{View}_{P}^{(\tilde{V}_3)}(x, s_3),$$

can be distinguished. The contradiction with divertibility follows.

The new combined prover $\tilde{P}^* = (\tilde{P}, W_1, \tilde{V}_1, W_2, \tilde{V}_2)$ is constructed as shown in Figure 3.7. $\tilde{P}^*$ can convince an honest verifier with overwhelming probability because of 2-divertibility. A sample of the random variable $\mathit{View}(x, s)$ will be denoted as $\mathit{view}(x, s)$ in the following.

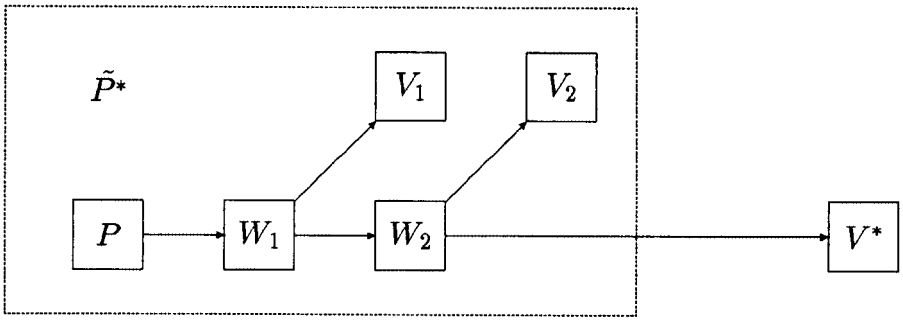

Figure 3.7: Constructing of $\tilde{P}^*$

When proving the claim to a verifier V^*, $\tilde{P}^*$ works as follows:

1. $\tilde{P}$ executes the protocol with W_1 who diverts it to $\tilde{V}_1$ and W_2, where $\tilde{P}$ simulates $W_1, \tilde{V}_1$, and W_2.

2. W_2 diverts the protocol to $\tilde{V}_2$ simulated by $\tilde{P}$, and a verifier V^*.

The view of $\tilde{P}^*$ includes the messages from V^* and the random bits.

Suppose M is an algorithm which distinguishes H_0 and H_1. The output i means that the input is from H_i, $i = 0, 1$. A distinguisher M' for H_{W_3} and H_V can be constructed as follows:

1. For any sample of H_{W_3} or H_V,

$$h = (view^{(\tilde{P}^*)}(x, s^*), view^{(\tilde{V}_3)}(x, s_3)),$$

   (a) from $view^{(\tilde{P}^*)}(x, s^*)$, produce a sample

$$h' = (view_{W_1}^{(\tilde{P})}(x, s), view_{W_1}^{(\tilde{V}_1)}(x, s_1), view_{W_2}^{(\tilde{V}_2)}(x, s_2),$$

   which can be done since $\tilde{P}^*$'s view includes all the random bits.

   (b) add view $view^{(\tilde{V}_3)}(x, s_3)$ to h' to get a sample

$$h'' = (view_{W_1}^{(\tilde{P})}(x, s), view_{W_1}^{(\tilde{V}_1)}(x, s_1), view_{W_2}^{(\tilde{V}_2)}(x, s_2), view^{(\tilde{V}_3)}(x, s_3),$$

   of H_0 or H_1

2. Input the sample h'' to M.

3. Get the output of M.

   If it is 0, then h'' is a sample of H_0. So h is a sample of H_{W_3}. Similarly, if the output of M is 1, h is a sample of H_V.

Since M can distinguish H_0 and H_1, H_{W_3} and H_V can be distinguished by M'.

Then the fact that H_1 and H_2 are indistinguishable can be proved by showing that if H_1 and H_2 are distinguishable then

$$H_W = (\, View_W^{(\tilde{P})}(x, s), View_W^{(\tilde{V}_1)}(x, s_1), View_W^{(\tilde{V}_2)}(x, s_2),$$

and

$$H_V = (\, View_W^{(\tilde{P})}(x, s), View_W^{(\tilde{V}_1)}(x, s_1), View_P^{(\tilde{V}_2)}(x, s_2),$$

are distinguishable, which is impossible for the same reason as that H_0 and H_1 are indistinguishable.

Suppose that M, with output 1 or 2, is a distinguisher of H_1 and H_2. An algorithm which distinguishes H_W and H_V is described as follows:

1. For any sample of H_W or H_V,

$$h = (view_W^{(\tilde{P})}(x, s^*), view_W^{(\tilde{V}_1)}(x, s_1), view^{(\tilde{V}_2)}(x, s_2)),$$

   $\tilde{P}$ produces a sample

$$h = (view_P^{(\tilde{V}_3)}(x, s_3),$$

   together with $\tilde{V}_3$.

2. Form a sample

$$h'' = (view_W^{(\tilde{P})}(x, s), view_W^{(\tilde{V}_1)}(x, s_1), view^{(\tilde{V}_2)}(x, s_2), view^{(\tilde{V}_3)}(x, s_3),$$

   as input of M.

3. Send h'' to M.

4. Get output of M.

   If it is 1 then h'' is a sample of H_1. h is recognized as a sample of H_W. Similarly, if the output of M is 2, h is a sample of H_V.

Finally, H_2 and H_3 are indistinguishable, since $\tilde{P}$ can produce the instances of both $View_P^{(\tilde{V}_2)}(x, s_2)$ and $View_P^{(\tilde{V}_3)}(x, s_3)$, as proving that H_1 and H_2 are indistinguishable, there exists an algorithm to distinguish

$$(View_{W_1}^{(\tilde{P})}(x, s), View_{W_1}^{(\tilde{V}_1)}(x, s_1),$$

and

$$(View_V^{(\tilde{P})}(x, s), View_P^{(\tilde{V}_1)}(x, s_1),$$

which is in contradiction with divertibility. □

We are now ready to state and prove the main result of this section.

Theorem 19 *If the protocol in Figure 2.1 is a witness hiding proof of knowledge then it is not* 2*-divertible.*

Proof As each element of E can be represented by a polynomial number of bits, the cardinality of E can be assumed to be at most $2^{q(k)}$ for some polynomial q. If it is 2-divertable, by Lemma 18 the proof is $q(k)$-divertible.

Let A_i denote the event that there is an $c \in E$ such that for more than $|E|/3$ of the possible challenges from the i'th verifier W can find an answer to this verifier given the prover's response to c. By Lemma 17 the probability of A_i is negligiblly small. Hence, the probability of the event $A = A_1 \vee \ldots \vee A_{q(k)}$ is superpolynomially small as well (over the choice of a, r and ρ). Let Acc denote the event that all verifiers accept (distributed according to the random coins of all parties). Then

$$\begin{aligned} Prob[Acc] &= Prob[Acc \mid A]Prob[A] + Prob[Acc \mid \neg A]Prob[\neg A] \\ &\leq Prob[A] + Prob[Acc \mid \neg A] \end{aligned}$$

Now if $\neg A$ occurs then a given c cannot be used to answer more than a third of the possible questions from the i'th verifier. Hence, the probability that there is a c which can be used to answer all verifiers is at most

$$|E|3^{-q(k)} \leq \left(\frac{2}{3}\right)^{q(k)}$$

This is also an upper bound on $Prob[Acc \mid \neg A]$. Thus the probability that all verifiers accept is negligible. Therefore, at least one verifier will not accept with overwhelming probability. □

3.4 Extensions

3.4.1 Transferability and divertibility with different inputs

In some divertible proofs the warden is not interested in proving that he knows a witness to x which is the common input of both (P, W^V) and (W^P, V), but rather to some transformation of x. The following example shows the possibility of transferring the protocol to a third party with different inputs.

Example Consider the example of transferring the semi-proof of square root modulo n appeared at the begining of this chapter, where the input is x (and n) for both (P, W^V) and (W^P, V). Now we suppose that the input for (W^P, V) is $x_1 = v_1^2 x$, where v_1 is chosen by W at random. W can transfer the protocol as shown in Figure 3.8.

Transferability and divertibility for different inputs can be generalized to commutative random self-reducible relation (see [OO89]). If it is possible to transfer or divert the protocol to a third party with different input, then all the proofs in the previous two sections work in this more general scenario. More precisely, we can prove the following:

Theorem 20 *Let E be of polynomial size, and the basic protocol be a semi-proof of knowledge with the input x. If it is 2-extractable, then any witness hiding proof of knowledge (P, V), in which the basic protocol is iterated t times satisfying $|E|^t$ grows faster than any polynomial in k, cannot be weakly 2-transfered by any polynomial time warden W to V_i with input x_i, $i = 1, 2$.*

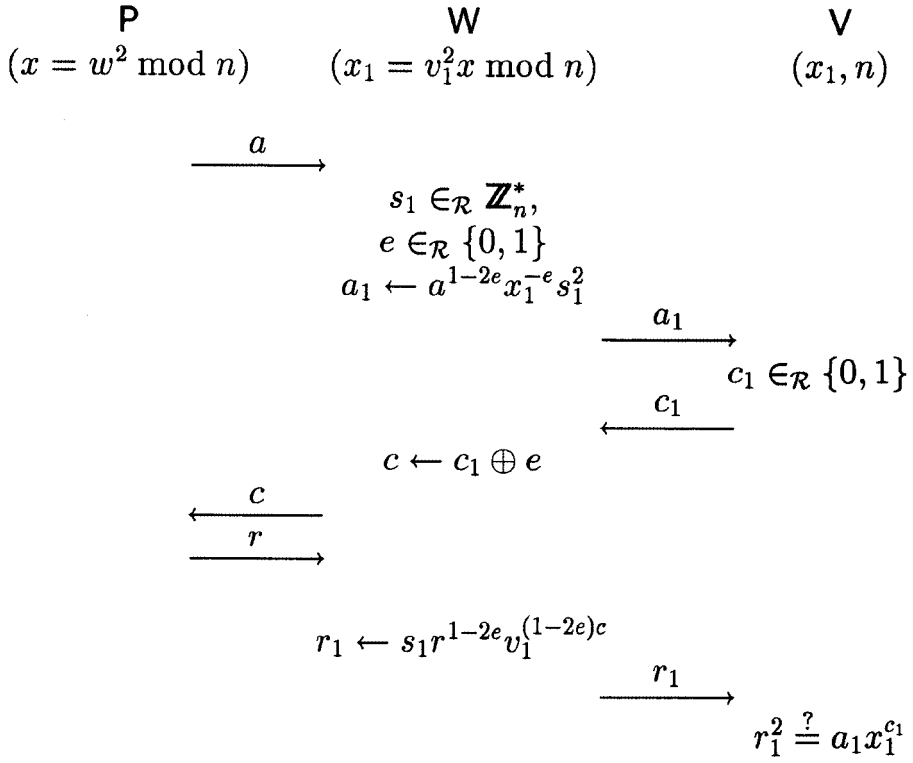

Figure 3.8: Diverting the semi-proof of square root with different input

Proof The method used in the proof of Theorem 15 is also suitable for different input transferability: If the warden can answer more than $\frac{1}{|E|}$ of the possible pairs of challenges from V_1 and V_2 with nonnegligible probability in

transferring the basic protocol to V_1 with input x_1 and V_2 with input x_2, then by constructing W' which transfers the protocol to V_1 or V_2 randomly, W' must be able to answer two challenges from V_1 or V_2 by using the same challenge to P with nonnegligible probability. Thus the witness of x_1 or x_2 can be extracted in polynomial time.

It can be reduced directly that if the basic protocol is iterated t times to form a witness hiding proof, where $|E|^t$ grows faster than any polynomial in k, then it is not weakly 2-transferable. □

The result for different input transferring can be extended to l-extractable witness hiding basic protocol without any difficulty.

Theorem 21 *Let E be of polynomial size, and the basic protocol be a witness hiding semi-proof of knowledge with the input x. If it is l-extractable, and if $(l-1)^n|E| < |E|^n$, then no warden can answer more than $(l-1)^n|E|$ of the possible tuples of challenges with non-negligible probability in n-transferring the basic protocol with inputs x_i to $V_i, i = 1, 2, \ldots, n$.*

Theorem 22 *If the basic protocol is a witness hiding proof of knowledge with input x, then it is not 2-adivertible with input x_1 and x_2 to V_1 and V_2 separately.*

Proof sketch: The key points of extending the proof of Theorem 19 to cope with different inputs are

- W cannot use the same c in order to answer too many challenges from V no matter for which input he is going to divert the protocol;
- if the proof can be diverted to two verifiers with inputs x_1 and x_2 separately then it can be diverted to n verifiers with inputs $x'_1, x'_2, \ldots, x'_n$ separately for any n, polynomial in k.

3.4.2 Divertibility of four move proof of membership

This section presents a divertible zero-knowledge proof of membership, in which both warden and verifier are convinced unconditionally. The original proof system was suggested by Chaum and used to verify undeniable signatures in [Ch91].

Let p and q be primes such that q divides $p-1$. The common input is $(g,h,m,z) \in \mathbb{Z}_p^*$ where g has order q. The prover knows $x = \log_g h$ and wants to show that $\log_g h = \log_m z$. The divertible proof system is shown Figure 3.9.

Theorem 23 *Even if P and V cooperate, the warden will not accept a false statement with probability larger than $(q-1)^{-1}$ (over his own coins).*

Proof In [Ch91] it is shown that if $\log_g h \neq \log_m z$ then the prover can only convince the warden if he can guess the value of a before sending h_1 and h_2.

Thus it is sufficient to show that if W chooses r, t'', a'', b'' uniformly at random then given $c', c, h_1, h_2, h_1', h_2', a', b'$ all values of a but one can ocurr (and with the same probability).

Given a pair (a,b) such that $c = g^a m^b$. Let $m = g^d$, $h_1 = g^{d_1}$ and $h_1' = g^{d_2}$. Similarly, let $z = h^e$, $h_2 = h^{e_1}$ and $h_2' = h^{e_2}$. Then all information about the secret choices of the warden is contained in the following four equations:

$$\begin{aligned}(d_2 + t'')r &= d_1 - (a'' + db'') \\ (e_2 + t'')r &= e_1 - (a'' + eb'') \\ a'' &= a - a'r \\ b'' &= b - b'r\end{aligned}$$

By subtracting the second from the first we get three linear equations in the unknowns r, a'', b'':

$$\begin{aligned}(d_2 - e_2)r &= d_1 - e_1 - (d-e)b'' \\ a'' &= a - a'r \\ b'' &= b - b'r\end{aligned}$$

This implies

$$(d_2 - e_2)r = d_1 - e_1 - (d-e)(b - b'r)$$

and thus

$$(d_2 - e_2 - (d-e)b')r = d_1 - e_1 - (d-e)b$$

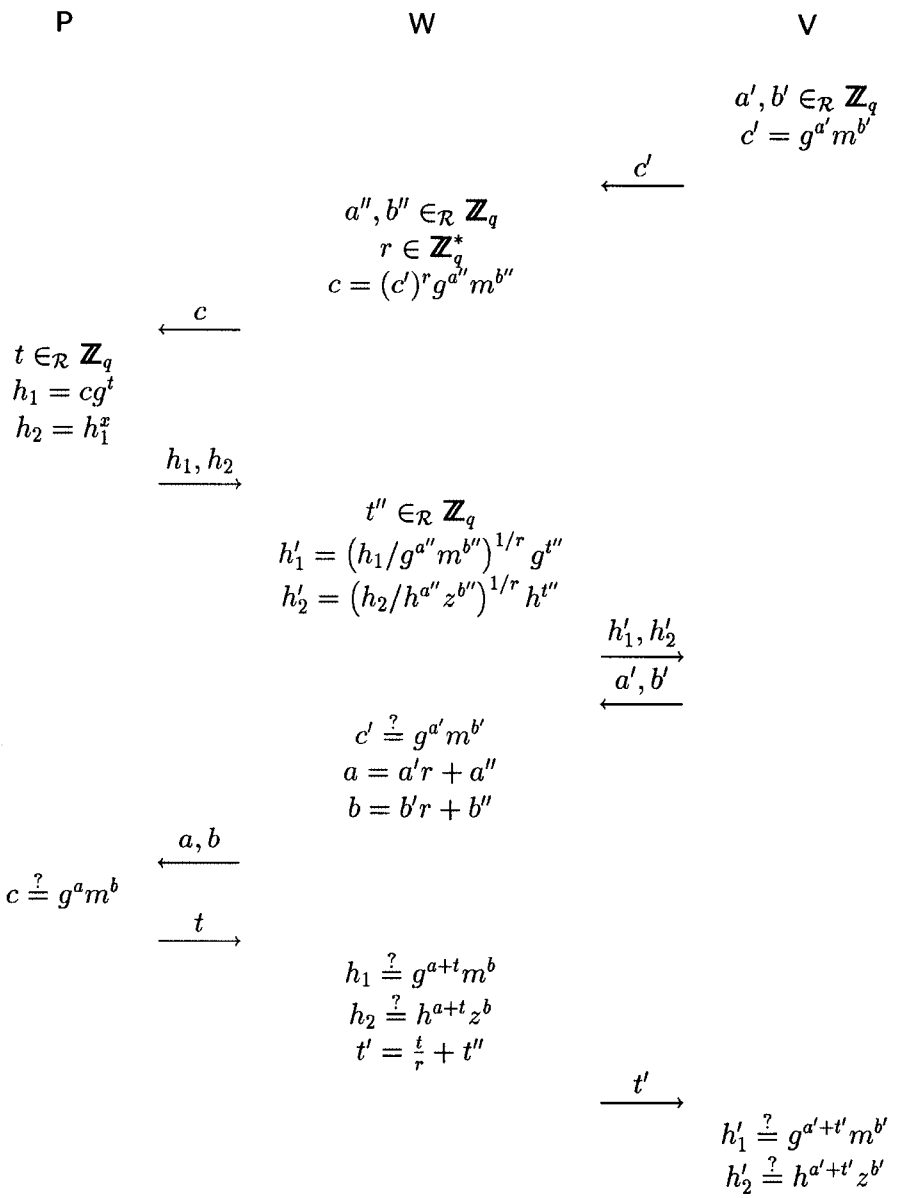

Figure 3.9: Convincing verifier and warden unconditionally.

We now distinguish two cases. First, if $d_2 - e_2 - (d-e)b' = 0$, then for any $r \in \mathbb{Z}_q^*, a'', b'', t''$ can be determined by

$$\begin{aligned}(d_2 + t'')r &= d_1 - (a'' + db'') \\ a'' &= a - a'r \\ b'' &= b - b'r\end{aligned}$$

It follows immediately, that also the equation

$$(e_2 + t'')r = e_1 - (a'' + eb'')$$

will be satisfied. Secondly, if $d_2 - e_2 - (d-e)b' \neq 0$, then r is determined by the above equation. As $r \neq 0$ the prover and verifier know that

$$b = \frac{d_1 - e_1}{d - e}$$

(and the corresponding value of a) is not possible. But for all other values t'', a'', b'' can be determined as in the first case.

As the prover chooses d_1 and e_1, he can always make sure that one of the q possible (a, b) pair will not occur. □

3.4.3 Blind signatures

In the first section, we have reviewed how to construct blind signatures based on divertible three move protocols which are witness hiding proofs of knowledge.

This kind of blind signature schemes is very difficult to analyse (e.g., its security depends very much on $\mathcal{H}$). However, it would be easy to get two signatures from one execution of the protocol, if the proof could be diverted to two verifiers in parallel. It follows immediately from Theorem 19, that such an attack cannot succeed with probability close to 1.

3.5 Conclusions and open problems

We have shown that witness hiding proofs based on iterating the basic three move protocol with polynomial size challenge set E cannot be transferred to

two independent verifiers simultaneously. If the basic protocol constitutes a witness hiding proof in itself, it cannot be diverted to two independent verifiers with overwhelming probability.

It would be interesting to improve the analysis of the latter case and obtain a result just as strong as in the former. Furthermore, it is an interesting open problem to extend the results in this paper to general proofs of knowledge (not only proofs based on the three move protocol). Especially, it will be significant to prove that it is impossible to divert the four move protocol which is a divertible zero-knowledge proof of membership in last section to two independent verifiers.

Chapter 4

Credentials with Pseudonyms

Credentials are certificates of a certain kind of personal information issued by some organization. Examples are a passport, a driver's license, a token for voting, a bank cheque, *etc.* In order to protect the privacy of individuals, these credentials are assigned to pseudonyms. To prevent creation of a record of the activities of an individual, the same person may use different pseudonyms in dealing with different organizations. In such a case, an individual should be able to transfer credentials issued to one pseudonym to another pseudonym in order to show it to some organization. Credential mechanism is used in literature to describe the whole process including establishing pseudonyms, issuing and transferring credentials (see [CE86]).

4.1 Main idea and basic protocol

The basic requirement for credentials is *unforgetability*, *i.e.* an individual cannot show a kind of credentials to an organization unless it has been properly issued to him. If the credentials are issued with pseudonyms, then they must satisfy *unlinkability*. This includes two aspects: no pseudonym can be linked to the identity of the individual, and two different pseudonyms cannot be found corresponding to one individual.

In order to protect the privacy of individuals, the pseudonyms must have some randomness. However, considering that credentials are a kind of certificates on the pseudonyms, then to prevent the individuals from abusing the system, all the pseudonyms must be formed according to some rules. So

the system must include a process which forces the individuals to form their pseudonyms in a right way but does not require individuals to reveal more about how they have actually constructed their. pseudonyms. This process is called validating pseudonyms.

Chaum and Evertse (see [CE86]) described an RSA based credential mechanism, where the credentials are RSA signatures on pseudonyms which are validated by cut-and-choose. Damgård [Da88] published another construction of a credential mechanism based on multiparty computing protocol with secret inputs and outputs. The credential system proposed here uses a discrete logarithm setting. The whole process, from validating pseudonyms to transferring credentials, is based on three move divertible proof of knowledge.

As in [CE86], [Da88], individuals are identified by pseudonyms. The participants are: a trusted center C, a set of organizations $\{O_j \mid j \in \mathcal{J}\}$, and a set of individuals $\{I_k \mid k \in \mathcal{K}\}$.

Suppose that p is a prime, and q is the largest prime factor of $p-1$, g is a generator of the multiplicative group G_q of order q. An individual with the physical identity ID_k (name, address, birth date, photo, *etc*) will be represented by I_k. C issues a personal identification number u_k to I_k. A pseudonym of I_k will then have the form

$$U_k = u_k g^r,$$

where r is chosen by I_k from $\mathbb{Z}_q^*$ randomly and secretly. The pseudonym U_k is independent of the physical identity ID_k. If U_k' is I_k's another pseudonym, then

$$U_k' = U_k g^s = u_k g^{r+s}$$

for some s which only I_k knows. So different pseudonyms are also independent of each other.

In a credential system, suppose the organization O_j issues credentials with public key (g, h) and corresponding secret key $\log_g h = x$. The credentials of the pseudonym U_k will be designed as

$$Z_k = U_k^x,$$

together with a proof that

$$\log_g h = \log_{U_k} Z_k,$$

by the following basic protocol shown in Figure 4.1 where U_k and Z_k are simply written as U and Z.

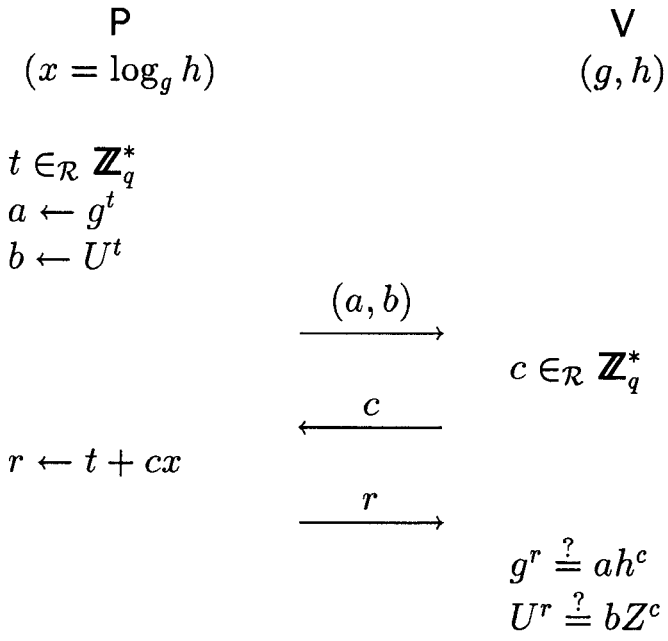

Figure 4.1: Basic protocol: proof of $\log_g h = \log_U Z$

This protocol was originally introduced by Chaum and Pedersen in [CP92]. It is a proof of knowledge of secret key x and $\log_g h = \log_U Z$. It can be proved to be witness hiding only when the challenge set is a subset A of $\mathbb{Z}_q$ and $|A|$ is polynomial of $|q|$. In [CP92], they conjectured that no matter which $c \in \mathbb{Z}_q$ is chosen as a challenge, the prover reveals no other information than the fact that $\log_g h = \log_U Z$.

Then from U_k^x, for his another pseudonym U_k', I_k can easily compute $Z_k' = U'^x_k = U_k^x h^s$ since he knows s. In order to prove the credentials on U_k' to another organization, the basic protocol is diverted with input (g, h, U_k', Z_k'). The divertibility is shown in Figure 4.2 where the index k's are also omitted.

The divertibility of the basic protocol is an online property. However, in most cases, credentials need to be shown to some organization afterwards. The blind signature scheme based on basic protocol will be introduced to show the credentials offline as in Figure 4.3.

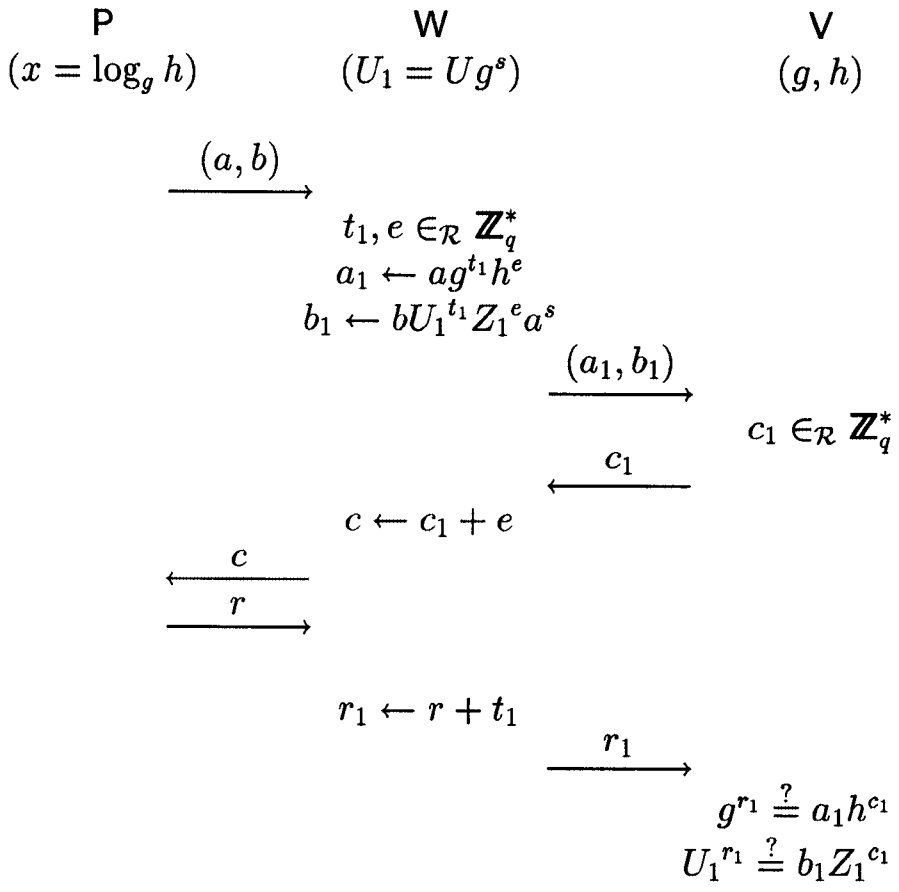

Figure 4.2: Diverting proof of $\log_g h = \log_U Z$

The signature based on basic protocol

Let $\mathcal{H}$ be a hash function (as in the Fiat-Shamir signature scheme, see [FS87]). Given this function and the basic protocol, the signature on U with secret key

$$x = \log_g h$$

is

$$\sigma_x(U) = (Z, a, b, r).$$

It is correct if $c = \mathcal{H}(U, Z, a, b)$ and

$$g^r = ah^c \quad \text{and} \quad U^r = bZ^c.$$

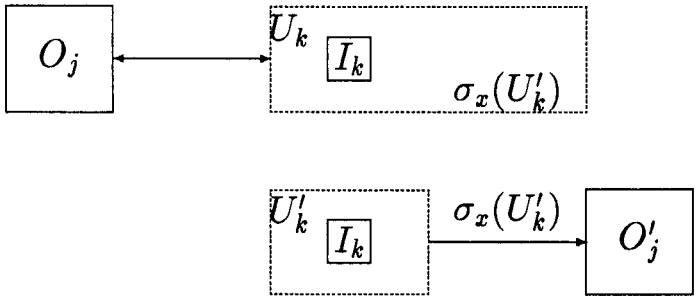

Figure 4.3: Transferring credentials

The security of this signature scheme depends on the property of the hash function $\mathcal{H}$ to a great extent. In this chapter, the following assumption is used.

Assumption 1 $\mathcal{H}$ has the property that if the basic three move protocol is a proof of knowledge, then it is as difficult to convince a verifier who chooses $c = \mathcal{H}(U, Z, a, b)$ as to convince a verifier who chooses c at random.

The blind signature

In order to get a signature on $U_1 = U\ g^s$, the warden W computes U_1^x from U^x. When executing the protocol with input (g, h, U, Z), he diverts the protocol with input (g, h, U_1, Z_1). But instead of getting a random challenge c_1 from verifier V, he computes $c_1 = \mathcal{H}(U_1, Z_1, a_1, b_1)$, and finally gets

$$\sigma_x(U_1) = (Z_1, a_1, b_1, r_1).$$

The example here will point the possible abuse of the individuals. If I_k and I_h cooperate together, say, I_k is to help I_h to pass driving examination, then they must find a way to transfer credentials issued to I_k's pseudonym U_k to I_h's pseudonym U_h. A possible way is that I_h borrows I_k's personal identification number u_k to form his pseudonym $U_h = u_k g^s$. To prevent this, the following combined protocol is used in validating pseudonyms.

The combined protocol

In this protocol, the public key is $(g^{\frac{1}{\beta}}, g, g^\alpha)$. The secret key for the prover is (α, β), where $\alpha, \beta \in \mathbb{Z}_q^*$. The common input is

$$(g^{\frac{1}{\beta}}, g, g^\alpha, U, A, Z_v).$$

The purpose of the protocol is to prove that

$$Z_v = U^\alpha A^{\alpha\beta}.$$

We combine the basic protocol as follows.

1. (P, V) execute basic protocol with input $(g^{\frac{1}{\beta}}, g, A, A^\beta)$.

2. (P, V) execute basic protocol with input $(g, g^\alpha, U\ A^\beta, Z_v)$.

3. V accepts if he accepts in both 1 and 2.

The combined signature

We define the combined signature on (A, U) employing the secret (α, β) as

$$\sigma_{(\alpha,\beta)}(A, U) = \sigma_\beta(A), \sigma_\alpha(U\ A^\beta)).$$

The diverted combined protocol

The combined basic protocol can be diverted to the protocol with input

$$(g^{\frac{1}{\beta}}, g, g^\alpha, U_1, A_1, Z_{v_1}).$$

where $U_1 = U\ g^r$ for some r and $A_1 = A(g^{\frac{1}{\beta}})^s$ for some s, $Z_{v_1} = U^\alpha A^{\alpha\beta} g^{\alpha(r+s)}$. It goes as follows:

1. The first step 1s diverted with $(g^{\frac{1}{\beta}}, g, A, A^\beta)$ as input of (P, W), and $(g^{\frac{1}{\beta}}, g, A_1, A_1^\beta)$ as the input of (W, V).

2. The second step is diverted in the same way as step 1. The input to (P, W) is $(g, g^\alpha, U\ A^\beta, (U\ A^\beta)^\alpha)$, and the input to (W, V) is $(g, g^\alpha, U_1 A_1^\beta, (U_1 A_1^\beta)^\alpha)$, where $U_1 A_1^\beta = U\ A^\beta g^{r+s}$.

Also blind combined signatures can be constructed by divertibility of combined protocol.

4.2 Validating pseudonyms

We assume the trusted center C works as a notary office to validate pseudonyms. It is trusted not to produce illegal pseudonyms for itself, just as a bank is trusted not to produce illegal money. The privacy of individuals does not depend on C.

For every individual I_k the center C chooses $u_k \in G_q$ as the personal identification number of I_k. It will be used as a factor of all his pseudonyms. It is reasonable to suppose that I_k does not know $\log_g u_k$ since u_k is chosen by C. We regard u_k as uniquely corresponding to I_k.

Before validating the pseudonyms for some organization O_j, C chooses (α, β), $\alpha, \beta \in \mathbb{Z}_q^*$ as secret key that will be used in the validating process, and publishes the corresponding public keys $g_1 = g^{\frac{1}{\beta}}, g, g_2 = g^{\alpha}$. The process of validating pseudonyms goes as follows.

1. I_k chooses $r_k \in \mathbb{Z}_q^*$ at random. He forms his pseudonym $U_k = u_k g^{r_k}$. Here U_k is represented as a number in $\mathbb{Z}_p$.

2. I_k computes

$$a_k = g_1^{U_k + s_k},$$

   where $s_k \in_R \mathbb{Z}_q^*$ and sends a_k to C.

3. C computes

$$u_k = u_k^{\alpha} a_k^{\alpha\beta},$$

   and sends it to I_k.

4. C proves to I_k that v_k is correct by executing combined protocol with input $(g_1, g, g_2, u_k, a_k, v_k)$. I_k gets the following combined blind signature as the validator for his pseudonym,

$$V_k = \sigma_{(\alpha,\beta)}(g_1^{U_k}, U_k) \qquad (*).$$

5. I_k sends U_k and V_k to O_j. If V_k is a correct validator of U_k, then U_k will be registered in O_j as a valid pseudonym.

If necessary, I_k will choose m_k as his secret key under the pseudonym U_k, and register $Q_k = U_k^{m_k}$ as public key. If it is necessary, Q_k can be validated in the same way as validating U_k.

The secret key (α, β) could be chosen by both C and O_j, depending on how much trust is placed in O_j. In some cases, the security will not rely on O_j. For example, in a voting scheme, the voting organization is not trusted, *i.e.* is presumed to be willing to forge votes. If it can produce valid pseudonyms, then it can forge votes. So we will prevent O_j from knowing (α, β).

In order to validate the pseudonyms for another organization $O_{j'}$, the same process will be repeated by using different key (α', β'). I_k's pseudonym in $O_{j'}$ is $U'_k, U'_k = u_k g^{r'_k}$.

After this process, each organization identifies a set of individuals by a list of valid pseudonyms, and each individual has different pseudonyms in several relevant organizations.

4.3 Issuing and transferring credentials

Each organization O_j is authorized to issue a particular type of credentials. For example, a police station may issue drivers' licenses.

Suppose that O_j issues the credentials with respect to the public key (g, h) and corresponding secret key $\log_g h = x$.

If the individual I_k is authenticated to get the credentials by O_j, where I_k is identified as U_k, then I_k gets

$$\sigma_x(U_k) = (Z_k, a, b, r),$$

where $Z_k = U_k^x$. It is correct if $c = \mathcal{H}(U_k, Z_k, a, b)$ and

$$g^r = a h^c \quad \text{and} \quad U_k^r = b Z_k^c$$

In order to show the credentials to organization $O_{j'}$, I_k must transfer the credentials to his pseudonym U'_k which identifies him in $O_{j'}$. Here

$$U'_k = U_k g^s$$

for some s which I_k knows. One way to transfer the credentials is the following: when I_k needs to show the credentials to O_j, he executes the basic

protocol with O_j playing the role of warden in order to get the credentials on U'_k. But this way is inconvenient, since every time I_k needs to show the credentials to some organization, he has to go to O_j and execute the basic protocol. Here a more convenient way is given to get several copies of the credentials, in order to be able to show to several organizations independently.

When I_k gets the credentials, he may have no idea which organizations he may have to show to afterwards. So copies shouldn't be customized for any predetermined organizations. One copy of the credentials of I_k is the signature

$$\sigma_x(U_k^*)$$

where $U_k^* = u_k g^t$, and t is chosen randomly by I_k. U_k^* is not necessarily any of I_k's pseudonyms. I_k can get it by executing the basic protocol with O_j.

When I_k needs to show the credentials to $O_{j'}$, he shows one of the copies $\sigma_x(U_k^*)$ and proves to $O_{j'}$ that he knows $\log_g(\frac{U'_k}{U_k^*})$ by executing the basic protocol with input (g, g^d, h, h^d) where $U'_k = U_k^* g^d$ is the pseudonym of I_k in $O_{j'}$.

Remark Different copies of the credentials are independent. But any given copy $\sigma_x(U_k^*)$ can only be used once, otherwise it links different pseudonyms.

For communication between the individuals and organizations, an anonymous channel is not assumed, because it is rather restrictive. Some transactions must be face to face, or I_k may find a "representative" as in [CE86]. Some documents, such as a driver's license or a health certificate, can only be issued after physical identifications.

For practical credential mechanisms, unlinkability cannot preclude revealing "strictly necessary" information (see [CE86]). If O_j is supposed to issue the credentials with secret key x, we define some sets for the organizations corresnonding to the credentials:

- for O_j, the set of the individuals who have got the credentials is denoted by $O(x)$;
- for $O_{j'}$ the set of individuals having shown the credentials with public key (g, h) is denoted by $O'(g, h)$.

The strictly necessary information is precisely described as the fact that any individual in $O'(g,h)$ is in $O(x)$. It is required that he is identified to a special individual in $O(x)$ with the probability $\frac{1}{|O(x)|}$. If $O(x)$ consists of just a single individual, then it is impossible to avoid linking the pseudonyms as a result of showing the credentials to some other organization $O_{j'}$.

In our system, the unlinkability holds in the following sense.

Definition 24 (unlinkability) Credential mechanism satisfies unlinkability in the sense that, for any individual I_k no more Shannon information than strictly necessary about the connection between either his identity and any of his pseudonyms, or between any of his two different pseudonyms, is revealed.

Theorem 25 (unlinkability) *The credential mechanism proposed satisfies unlinkability, even if the center and wme organizations cooperate.*

Proof The validator of a pseudonym is a blind signature which is independent of u_k. So it does not reveal any connection between u_k and pseudonym. Transferring credentials from one pseudonym to another by blind signature will not link those pseudonyms either. □

4.4 Unforgeability of credentials

Consider the credentials with public key (g,h) and secret key $x = \log_g h$. For any I_k, forging one copy of credentials is equivalent to forging the credentials of one of his pseudonyms. So we treat only forging credentials of pseudonyms in this section.

Definition 26 (unforgeability) The credential mechanism satisfies unforgeability, if no individual can forge the credentials with any of his valid pseudonym U_k after a cooperating group of individuals obtain the credentials.

The unforgeability of our scheme depends on the discrete logarithm assumption and the following conjecture.

Conjecture 1 *For any polynomial time warden W, if the basic protocol with input (g, h, U, Z) can be diverted by W for input (g, h, U_1, Z_1) to V, then either W knows the secret key $x = \log_g h$, or $U_1 = U^i g^s$ for some $i, s, i \neq 0$, that W knows.*

This conjecture is assumed to be true in [Bran94a]. Some arguments have been presented in [Bran94b], which suggest that breaking it requires breaking either the Schnorr scheme (see [Sch90]) or Diffie-Hellman assumption (see [DH76]).

Theorem 27 (unforgeability) *The credentials described above satisfies unforgeability under the discrete logarithm assumption, if the conjecture is true and the hash function in the signature scheme has the property stated in Assumption 1.*

Proof By Assumption 1, it is infeasible for any I_k to compute the credentials by himself without knowing the secret key x, no matter what U_k is after sufficiently many cooperating individuals have obtained the credentials.

If I_k can transfer credentials issued to somebody else, say $I_h, k \neq h$, with pseudonym U_h, to his own pseudonym U_k, then by the conjecture,

$$U_k = U_h^i g^s$$

for some $i, s, i \neq 0$, known by I_k. We will prove that this is infeasible even if I_h cooperates with I_k.

By using the conjecture twice, we can establish that the pseudonym U_k of I_k with personal identification number u_k can be validated, if and only if

$$U_k = u_k^{\lambda_1} g^{\lambda_2},$$

where I_k knows λ_1 and λ_2, $\lambda_1 \neq 0$. Similarly, if U_h is I_h's valid pseudonym then

$$U_h = u_h^{\tau_1} g^{\tau_2},$$

and I_k knows τ_1 and τ_2, $\tau_1 m \neq 0$.

If for some i, s, the equality $U_k = U_h^i g^s$ holds, then I_k can compute ν_1 and ν_2 such that

$$u_k = u_h^{\nu_1} g^{\nu_2},$$

which contradicts the discrete logarithm assumption. □

In our credential mechanism, we suppose that different credentials are issued with different public (secret) keys which are chosen independently. The cooperation of some organization $O_{j'}$ with certain individuals does not help forging the credentials which are supposed to be issued by O_j. If it does, $O_{j'}$ and the individuals together must hold the secret key chosen by O_j, since the credentials are based on a proof of knowledge. If an organization issues several different credentials, it must choose the corresponding secret keys independently and randomly in order to make the credentials secure.

4.5 Discussions

The credential system proposed in this chapter is provably secure under the discrete logarithm assumption if Conjecture 1 is true.

Shifting the credential system from an RSA setting to a discrete logrithm setting simplifies the process of validating pseudonyms by droping cut-and -choose. The model in this chapter is closer to Damgård's model (see [Da88]) in the sense that the center will not be needed any more after the pseudonyms are validated, since each organization has its own secret key for issuing credentials without requiring the center to implemente computing task. Furthermore, the individuals can validate their own secret keys which will be used when the signatures are necessary under the pseudonyms.

But one notable special property of this credential system is that a copy of the credentials can be shown only once. If an individual wishes to show the credentials to a second organization $O_{j''}$ after showing to $O_{j'}$, he must get another copy from O_j. This property is suitable for one-time credential (see [CP92]), such as a bank cheque. For other purposes, this property does not seem convenient. It is an open problem to construct credential system based on discrete logarithms in which the credentials, once issued, can be shown in many different organizations independently.

Proving Conjecture 1 is another interesting open problem.

Chapter 5

Practical Elections

5.1 Introduction

In democratic societies, election is always executed around a locked box with a narrow opening on its top where all voters put their ballots in one by one. The purpose of this procedure is to ensure the anonymity of votes.

The development of electronic communication technologies brought about a dramatic increase in the volume and speed of information processing and distribution. Computers and networks replaced paper media in many transactions. Naturally, electronic voting is also feasible.

Theoretically, an election scheme is a multiparty computation with secret input values such that the correctness of the output is verifiable. The issue of multiparty computations has generated a number of articles (see [CCD88], [CDG88]).

Practically, election can be realized by encryption (see [Ch81]), blind signature (see [FOO92]), and other cryptography techniques. In order to avoid disruption, secret keys are shared in different ways. Many properties were claimed under assumption of some trusted special participants in the literature, which is necessary, since in practical election, voters should not be assigned too much computation. Two typical election models will be reviewed as a basic starting point to introduce a formal definition of practical election.

Several basic properties for election have been stated in the literature (see [FOO92]). But they are not close to formal definitions. The definitions given in this chapter are practical election models and relevant properties:

completeness, soundness, robustness, and privacy.

Two new practical elections are proposed. In the first, each eligible voter will be certified by credentials issued with pseudonyms as constructed in Chapter 4. Voters vote under their pseudonyms. The ballots can be signed by voters with validated public keys under their pseudonyms.

In the second election scheme, a "voting coin" is constructed to certify the voters, which is based on electronic cash. The voting coin can be used, not only as a token, but also as a ballot in any one of a variety of kinds of election. Such ballots cannot be altered after they have been casted.

5.2 Two typical practical election models

In this section, two models are reviewed. They are considered as the typical applications of cryptography in election.

5.2.1 Hide voters' identities

The first and most natural design for an election model is a mix network called mix (see [Ch81]) which can be considered a direct electronic variety of the locked box in conventional election. The mix is to scramble messages in order to protect the identities of the senders. Before being sent, each message is probabilistic encrypted (see [GM84a], [GM84b]). The messages are divided into different batches with a special size. The mix decrypts messages and outputs them per batch in a lexicographical order. Figure 5.1 shows how a single mix works, where E is an encryption algorithm, r_i's are random numbers, and m_i's are messages.

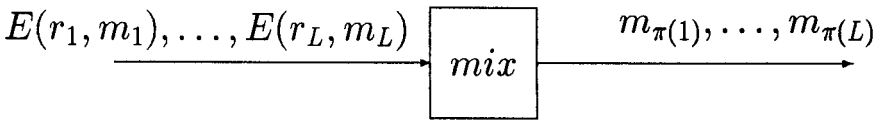

Figure 5.1: How a single mix works

In order to enhance the security of the system, several mixes are cascaded as shown in Figure 5.2 (see [Ch81]). In this case, messages are encrypted by

several different keys. A batch of messages $m_1, m_2, \ldots, m_L$ are encrypted as

$$\{E_1(r_i^{(1)}, E_2(r_i^2, \ldots, E_n(r_i^{(n)}, m_i)\ldots))\}_{i=1}^{L}$$

Each mix decrypts one encryption and outputs the results in lexico-graphical order.

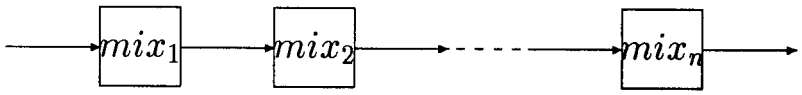

Figure 5.2: Cascaded mixes

The voting scheme proposed in [Ch81] uses the mix network for sending encrypted ballots. It is clear that the privacy depends both on the size of the batches and the security of encryption scheme. Park *et al.* proposed a mix network recently based on ElGamal public key system (see [PIK93]). The mix network is a kind of an anonymous channel. In [PIK93], a definition of a secure anonymous channel is given, which includes completeness, privacy, and verifiability.

In a practical voting scheme, all eligible voters must be certificated, before they send their ballots to the anonymous channel. As in a conventional election, each eligible voter will get a token or a stamped blank ballot which are necessarily untraceable. This problem, in fact, has been solved in electronic cash systems (see [CFN90]). Iversen proposed a voting scheme with untraceable tokens which are constructed along the same principle as electronic cash (see [Iv92]. Later, Fujioka *et al.* constructed a voting scheme (see [FOO92]) by using blind signatures of an authority to encrypted (commited) votes such that the channel only accept legal voters' votes by verifying the signatures. But the authority is assumed untrusted in the sense that it may forge illegal votes or may delete legal votes without counting them. So the restriction is added that eligible voters are not allowed to abstain, since otherwise, the authority can frame abstained voters without being caught and the disputation will produce when the votes are short for not being able to distinguish that the voters abstain or the authority deletes some votes.

5.2.2 Hide voters' votes

Another distinguished work on election schemes based on cryptographic technology, was proposed by Cohen (alias:Benaloh) and Fisher (see [CF85]). In

their voting scheme, instead of hiding the identities of voters, it hides the voters' votes. Each blank ballot is encrypted probabilistically in the sense that different votes, say "yes" and "no", cannot be distinguished with limited computational power. The communication is published via some media, called bulletin boards, which can be thought of as restricted shared memories. Each participant controls one bulletin board. The correspondence between bulletin boards and participants is fixed in advance. Each bulletin board can be read by every participant, but only its owner has write-access. Each voter encrypts all the possible votes and writes them on the board, which is called a blank ballot. The voter must prove that her blank ballot is encrypted in a correct manner. She will then mark one as shown in Figure 5.3.

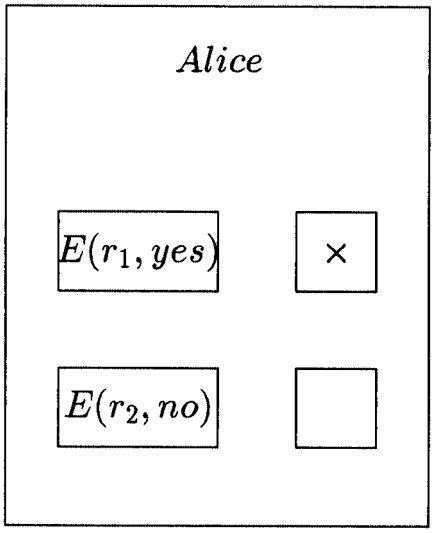

Figure 5.3: An encrypted ballot (on bulletin board)

A government, in the notation of [CF85], holding the key for decrypting the votes, counts the result from encrypted votes and proves that the tally is a correct one. In this case, it is efficient to protect the voting from being abused by both voters and government. The scheme achieved the desirable robustness and verifiability (see below for formal definitions). However, the privacy is not so strong as to prevent the government from knowing the correspondence between voters and ballots. The scheme was improved by Benaloh and Yung (see [BY86]) by sharing the decrypting key among several governors such that the privacy is prevented from any subset of government

members.

The two types of election schemes, which either hides voters' identities by mixes or hides votes by encryption are equivalent in the sense that in the former it shares the decrypting keys among different mixes, and the later, among different governors.

In next section, a general election model will be defined which includes the schemes mentioned above as special examples.

5.3 General election model

An election is a multiparty protocol. The participants are: a set of voters $\mathcal{V} = \{V_i\}_{i=1}^{N}$, and a set of governors $\mathcal{G} = \{O_t\}_{t=1}^{l}$. The common input for all participants is denoted as X, which includes a time limitation $[t_0, t_n]$. It should be understood as real time. An input which is sent in this time period is valid. A security parameter k is defined as the length of input.

An outsider M is allowed who is not necessarily to join the protocol. But it is possible for him to input some "rubbish" as an additional input of the protocol denoted as $\tilde{v}$.

A set called a secret input set for voters is defined as

$$\mathcal{X} = \{x_1, x_2, \ldots, x_m\}.$$

Let v_i denote the secret input of V_i.

For any inputs of voters and outsider $(v_1, v_2, \ldots, v_N, \tilde{v})$, a counter function is defined as

$$f_c(v_1, v_2, \ldots, v_N, \tilde{v}) = (c_1, c_2, \ldots, c_m)$$

such that

$$c_j = |\{v_i \mid v_i = x_j, 1 \leq i \leq N\}| \quad j = 1, 2, \ldots, m.$$

v_i is counted if and only if $v_i \in \mathcal{X}$. In particular, $\tilde{v}$ will not be counted no matter what it is.

For any number of voters N, an output T of the protocol is said to be *acceptable*, if

$$T = (c_1, c_2, \ldots, c_m),$$

where c_j's are nonnegative integers such that

$$\sum_{j=1}^{m} c_j \leq N.$$

The set of acceptable output will be denoted as $\mathcal{T}_N$.

For an election, some properties are defined as follows.

An election should have the property that all a voter can do to influence the total is to cast a single valid vote.

Definition 28 (Completeness) An election satisfies completeness, if for any $c > 0$, there exists k_0, such that when $k > k_0$,

$$Prob[T = f_c(v_1, v_2, \ldots, v_N, \tilde{v})] > 1 - \frac{1}{k^c},$$

if all participants follow the protocol. The probability is over all the participants' random bits.

Soundness of an election means that a falsified result cannot be produced as a valid output. In the literature, it is called verifiability (mentioned before) which means all participants can verify the outcome of the protocol. If we consider the output of the protocol as a result which has been passed the verifying of the voters, then soundness means that an acceptable output must be a correct result.

Definition 29 (Soundness) An election satisfies the soundness property, if for any $c > 0$, there exists k_0 such that when $k > k_0$,

$$Prob[T \neq f_c(v_1, v_2, \ldots, v_N, \tilde{v}) \wedge T \in \mathcal{T}] < \frac{1}{k^c},$$

The probability is over all the participants' random bits.

For an election, a weaker definition for soundness must be considered. Suppose that $\mathcal{G}^{C_1}$ is a set of subsets of $\mathcal{G}$,

$$\mathcal{G}^{C_1} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_r \mid \mathcal{G}_i \subseteq \mathcal{G}\}$$

Definition 30 (Weak-soundness) An election satisfies the weak soundness property relative to $\mathcal{G}^{C_1}$, if it satisfies soundness under the assumption that at least the governors in one of the $\mathcal{G}_i$'s follow the protocol.

An election is robust if the voters are unable to disrupt the protocol by sending messages (whether valid or invalid). It does not matter if the protocol identifies the disrupter, but it must come to a proper ending.

Definition 31 (Robustness) An election satisfies the robustness condition, if for any v_i, $i = 1, 2, \ldots, N$, and $\tilde{v}$, for any $c > 0$, there exists k_0 such that when $k > k_0$,

$$Prob[T \in \mathcal{T}_N] > 1 - \frac{1}{k^c}.$$

The probability is over all the participants' random bits.

In a voting scheme, any set of colluding participants can compute the total of all other voters by comparing their total to the total of all voters. The privacy protection of voting scheme must be compared to the information that can be obtained by collusion. In the ideal case, this is all the information that a colluder gets. Also the privacy is not always preserved against all the governors, but a proper subset of them. For example, in Benaloh and Yung's model, the secret key is shared among n governors. If all of them cooperate together, then they can find out what a voter has voted.

Suppose that $\mathcal{C}$ is a subset of $\mathcal{V}$, for simplicity,

$$\mathcal{C} = \{V_1, V_2, \ldots, V_{N_0}\},\ N_0 < N.$$

Let $\mathcal{H} = \mathcal{V} - \mathcal{C}$. Then $h_T = (v_{N_0+1}, v_{N_0+2}, \ldots, v_N)$ is called an *assignment* of $\mathcal{H}$ and

$$f_c(\phi, \ldots, \phi, v_{N_0+1}, \ldots, v_N, \tilde{v}) = T_{\mathcal{H}}$$

is called a *subtally* of $\mathcal{H}$.

Suppose h_0 and h_1 are two assignments of $\mathcal{H}$ with the same subtally $T_{\mathcal{H}}$. For each $e \in \{0, 1\}$, define an election $\mathcal{E}_e(\mathcal{C})$ as follows:

1. The voters in $\mathcal{H}$ follow the protocol with secret inputs as the assignment h_e.

2. The voters in $\mathcal{C}$ execute the protocol with any secret inputs.

For any subset $\mathcal{G}_i \subseteq \mathcal{G}$, any $\mathcal{C}$, and any participant $P \in \mathcal{C} \cup \mathcal{G}_i$, $\mathcal{C} \cup \mathcal{G}_i$ is a *conspiracy* with P, if it is permitted to communicate privately among $\mathcal{C} \cup \mathcal{G}_i$ during executing the protocol and P produces an output in $\{0, 1\}$. The probability of outputing 1 by executing a random run $\mathcal{E}_e(\mathcal{C})$ will be denoted as p_e.

Definition 32 (Privacy) Suppose $\mathcal{G}^{C_2}$ is a set of subsets of $\mathcal{G}$. An election preserves privacy relative to $\mathcal{G}^{C_2}$, if for any $\mathcal{C} \subseteq \mathcal{V}$, any $\mathcal{G}_i \in \mathcal{G}^{C_2}$, the conspiracy $\mathcal{C} \cup \mathcal{G}_i$ with any P has the property that for any two assignments with the same subtally, h_0 and h_1 of $\mathcal{H} = \mathcal{V} - \mathcal{C}$, and for any c, there exists k_0, such that when $k > k_0$,

$$|p_0 - p_1| < \frac{1}{k^c}.$$

Under this model, if the voting scheme employs a anonymous channel which is a mix network consisting of n mixes as in Figure 5.2, and O_i is a holder of i-th secret key, $i = 1, 2, \ldots, n$, then $O_1, O_2, \ldots, O_n \in \mathcal{G}$. If the public key system used is secure (see definition in [NY90]), and there is only one batch, then the voting scheme preserves privacy relative to $\mathcal{G}^{C_2}$, where $\mathcal{G}^{C_2}$ includes all the true subsets of $\{O_1, O_2, \ldots, O_n\}$.

Similarly, if the voting scheme uses encrypted ballots as in Benaloh's model and the secret key is shared among n government members $O_i, i = 1, 2, \ldots, n$, then it will preserve privacy relative to $\mathcal{G}^{C_2}$, which includes all true subsets of $\{O_1, O_2, \ldots, O_n\}$

In this sense, the two different types of voting schemes reviewed in Section 5.1 are equivalent. In order to make an election satisfy completeness, (weak) soundness, robustness, and privacy conditions, the secret key might be shared among $\mathcal{G}$ in different ways. The secret key can also be shared among voters. However for a practical voting scheme, the voters cannot be

assigned too much computation. In next two sections, two voting schemes will be presented which adopt a mix network as the anonymous channel.

The schemes consisting of the following three main stages will be considered:

- **Voter certification:** each eligible voter gets a token from $\mathcal{G}_1 \subseteq \mathcal{G}$, a subset of governors.
- **Voting:** voters send their ballots to an anonymous channel which is managed by $\mathcal{G}_2 \subseteq \mathcal{G}$.
- **Counting votes:** a subset of governors, $\mathcal{G}_3 \subseteq \mathcal{G}$, count the votes and publish the result.

5.4 Voting under pseudonyms

In the previous chapter, a credential system with pseudonyms has been established. In this section, the token, as credentials, is issued on personal identification numbers which can be transfered to pseudonyms in an untraceable manner. Suppose that O_a is a pseudonym notary office where the voters will validate their pseudonyms and O_b is a voting center which will issue tokens to eligible voters and count votes. The same schemes as in Chapter 4 will be used in validating pseudonyms, issuing and transferring credentials.

For a voter V_i with personal identification number u_i, his pseudonym is U_i and public key is (U_i, Q_i), where $U_i = u_i g^{r_i}$ and $Q_i = U_i^{m_i}$. V_i chooses r_i and m_i randomly and secretly. Both U_i and Q_i are validated by O_a as described in Chapter 4. The validators are F_i and E_i respectively.

A token, $\sigma_x(U_i)$, is a signature on U_i with secret key $x = \log_g h$. It is issued by Q_b as credentials with the personal identification number u_i of V_i and can be transfered to U_i.

In the voting stage, each voter sends

$$(U_i, Q_i, F_i, E_i, \sigma_x(U_i), v_i, \sigma_{m_i}(v_i))$$

to O_b by an anonymous channel, where $\sigma_{m_i}(v_i)$ is V_i's signature on vote v_i with his secret key m_i. O_b will publish the list shown in Table 5.1 to all voters.

In the stage of counting votes, v_i is counted by O_b if and only if

Pseudonyms	Tokens	Votes	Signatures
U_i, Q_i, E_i, F_i	$\sigma_x(U_i)$	v_i	$\sigma_{m_i}(v_i)$
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

Table 5.1: Voting list

- U_i only votes once;
- both U_i and Q_i are validated;
- $\sigma_x(U_i)$ is a correct token;
- $\sigma_{m_i}(v_i)$ is a correct signature with the public key (U_i, Q_i).

The properties for the election proposed above will be discussed separately. We will not give formal proofs under the definitions given in Section 5.3. For practical schemes, they might be effected by many very complicated factors. For example, if a practical voting scheme is executed in several stages, and each stage can be proved having some properties, it is hard to conclude that the scheme has the propertiese. In one of the references, it was claimed that several properties of the voting scheme have been proved (see [FOO92]). But they are far away from formal proofs. It is necessary to give formal definitions for the voting scheme and some properties which the scheme should satisfy. For the practical scheme, it must be considered in which sense it satisfies the defined properties. It is just the way in which practical elections are investigated in this chapter.

Completeness: In this scheme, if all the participants follow the protocol, then whether the result of a voting is correct depends on which kind of an additional input an outsider can produce. It has been proved in Section 4.4 that the credential mechanism is provably secure. If all the participants follow the protocol, the outsider cannot produce $\tilde{v}$ under a valid pseudonym and a correct token as an additional input. So the result will be counted correctly. The completeness is satisfied.

Robustness: Even though voters and an outsider can input v_i's or $\tilde{v}$ in some strange values, they cannot prevent O_b from producing an acceptable tally. Note that in the definition of robustness, the way in which the voters and the outsider to disrupt the voting is restricted to sending messages.

Privacy: Suppose that $O_1, O_2, \ldots, O_n$ are secret key holders for anonymous channel. $\mathcal{G} = \{O_a, O_b, O_1, \ldots, O_n\}$ in the election. Since both pseudonym validating and credential transferring are unlinkable, the privacy is preserved relative to $\{O_a, O_b\}$. Define

$$\mathcal{G}^{C_2} = \{\{O_a, O_b\} \cup \mathcal{G}_i \mid \mathcal{G}_i \subseteq \{O_1, \ldots, O_n\}, |\mathcal{G}_i| < n\}.$$

The election satisfies privacy relative to $\mathcal{G}^{C_2}$.

Soundness: For the soundness, we must consider how the voting result can be changed without being detected. Suppose that the governors $O_1, O_2, \ldots, O_n$ who manage the mix network are honest in the sense that each of them decrypts the inputs correctly and outputs all the messages without adding, changing and deleting. Then the result can be changed in two ways: by adding votes in for abstained voters and by deleting legal votes. A token is valid if and only if it is generated on a valid pseudonym (validated by a special kind of signature: combined signature) and is a correct signature relative to the public key. The key for validating pseudonyms is held by O_a, and the secret key for tokens is held by O_b. If one of O_a and O_b is honest then the tokens cannot be forged. But if O_b is not honest, he can delete legal votes. So the scheme satisfies soundness relative to

$$\mathcal{G}^{C_1} = \{\mathcal{G}_1 = \{O_b, O_1, \ldots, O_n\}\}.$$

5.5 Electronic cash based voting scheme

By shifting the characteristic properties of a voting scheme drastically towards a more realistic situation, a whole new tool becomes available. Both electronic coins and tokens for voting are credentials. And both of them must be untraceable. If we consider a coin to be a token for voting, then we can draw a parallel between cash systems and voting systems.

But for electronic voting schemes, in order to prevent from an alteration of the votes, the token and vote must be inseparable. Otherwise, for example,

under a valid token, if a "yes" vote can be changed to "no" vote, then the election will not satisfy soundness. The idea of using electronic cash as a token has been presented in [Iv92]. But in that case, the token is simply electronic cash which has no function to prevent the counter from falsifying the votes.

In this section, "voting coin" will be constructed by using known electronic cash systems. By the voting coin, the voter can vote in a variety of ways which can not be falsified afterwards.

5.5.1 Voting coins

Almost all known electronic cash systems can be used to construct voting coins. We start with an example.

At CRYPTO'93, Brands proposed an electronic cash system which is based on the representation problem in groups of prime order [Bran94a]. In this section, we use Brands' cash to construct a voting coin.

Assume G_q is a group of order q, a prime. Suppose that g_1 and g_2 are generators of G_q. A pair of generators (g, h) is the public key of the bank and $x = \log_g h$ is the secret key.

The user U has the account number

$$I = g_1^{u_1}$$

in the bank with random number u_1 chosen by U and kept secret from the bank.

When the user withdraws money from the bank, the message

$$m = Ig_2$$

is formed, and blinded as

$$m' = m^s,$$

for randomly chosen s by U. One coin from the bank is the blind signature on m' with the secret key x. Assume that H is a hash function satisfying Assumption 1. The blind signature is based on the three move diverted extension of basic protocol in previous chapter, and has the form

$$C = \sigma_x(m') = (A, B, z', a', b', r')$$

such that $c' = \mathcal{H}(A, B, z', a', b')$, $g^{r'} = a'h^{c'}$ and $m'^{r'} = b'z'^{c'}$, $AB = m'$. Here the user knows $A = g_1^{x_1}g_2^{x_2}$ and $B = g_1^{y_1}g_2^{y_2}$, the representations of A and B with respect to (g_1, g_2).

If the user U spends C in a shop S, S will first check if the coin is a correct signature from the bank. Then he will choose a challenge $u \in \mathbb{Z}_q^* - \{1\}$. U's reply to v is

$$(z_1, z_2)$$

where $z_i = x_i + vy_i$, $i = 1, 2$. S will check whether

$$g_1^{z_1}g_2^{z_2} = AB^v$$

If U spends C in another shop S' then, with high probability, S' will choose a challenge $v' \neq v$. From the replies to v and v', U will be traced (see [Bran94a]).

Now, instead of spending the coin C, the voter uses C to vote. We assume that the set of possible votes is a subset of $v \in \mathbb{Z}_q^* - \{1\}$. The vote is

$$(C, z_1, z_2, v).$$

When the votes are counted, the center will check if C is a valid coin and the equation $g_1^{z_1}g_2^{z_2} = AB^v$ holds.

No one can change a vote value v under C to another value v' unless he can get a representation of $AB^{v'}$, $v \neq v'$. But this is assumed to be infeasible. A voter cannot vote twice by using the same coin if there is a counting rule that if C repeat more than once with the same value v, then thev will be counted only once. If the voter votes different values by a coin, then he will be identified by the property of the cash.

Remark In order to prevent the bank which works as a voting center here from framing the voters, the account number could be $I = g_1^{u_1}g_2^{u_2}$ for generators g_1 and g_2 of G_q. So even with unlimited computational power the bank cannot frame the voters.

Most known electronic cash systems have the property that if a unit of cash is spent twice, then the user can be traced. This property can be implemented in the payment process by the random challenge c of a shop.

Since different shops will choose different challenges with high probability, the double-spender has to reply to two different challenges. If he does so, his identity will be revealed (see [CFN90]).

We use the challenge set as the set of votes. Instead of the shop choosing the challenge, the voter (user) chooses it as a vote value. Generally, even the bank cannot reply to a challenge without knowing the user's random choice for this coin. So no one can change the value of a vote. And a voter cannot vote more than once without having his identity revealed.

We notice that most electronic cashes are realized by blind signatures. A voter's choice of a vote value does not provide any benefit in forging the blind signature (cash).

5.5.2 General coin-based voting

Suppose that the bank is a trusted center O_a that issues voting tokens (coins) to eligible voters such that every eligible voter can get one and only one voting coin. The bank is said to be honest if it will not produce illegal voting coins for itself.

The voting organization is a counter O_b. It collects and counts the votes. The counter is honest, if it will not delete the valid votes.

In this voting model, it is not necessary to suppose on-line checking. The voter can repeat his vote (but not huge number of times which the counter cannot manage). According to the rule of counting votes, all the same votes by one token will be counted once. If a voter votes different values by one token, then he can be traced.

If n cascaded mixes will be used as an anonymous channel with $O_1, O_2, \ldots, O_n$ as secret key holders, then the privacy is preserved relative to any $\mathcal{G}_i \cup \{O_a, O_b\}$ where $\mathcal{G}_i$ is a true subset of $\{O_1, O_2, \ldots, O_n\}$.

It satisfies soundness relative to $\mathcal{G} = \{O_a, O_b, O_1, \ldots, O_n\}$.

5.6 Discussion and further work

In order to design a practical election which satisfies completeness, soundness, robustness and privacy, the secret keys are shared among governors in different ways. For most known work, whether or not an election scheme can

have the proposed properties depends on the way of sharing power. Some physical devices have been introduced. For example, bulletin boards (see [BY86]) are supposed to have the property that the messages written on them can be neither deleted nor altered. So it is impossible for the counter to delete the votes without being detected. Another practical way to prevent the counter from deleting the votes is to install a tample-resistant device at the voter's equipment. Each time a vote is received, the counter could record a receipt in the device. The voter can verify it immediately after he casts the vote. From the list, the voter can perceive the cheating if his vote is not counted. The receipt will serve to settle any dispute.

For a practical election, there is a limit as to how much computation and interaction voters can execute. But there is no clear definition of efficiency.

The most interesting problem is to construct an anonymous channel, which is fundamental for a practical election, such that it is provably secure.

Chapter 6

Group Signatures

6.1 Introduction

Group signatures as introduced in [CvH91] are signatures with the following three properties:

- only members of the group can sign messages on behalf of the group;
- the recipient of the signature can verify that it is a valid signature of that group, but cannot discover which member of the group created it.
- in case of dispute later on, the signer can be identified with the help of group members or a trusted authority.

Such a signature scheme can for example be used in invitations to submit tenders. All companies submitting a tender then form a group and each company signs its tender anonymously using the group signature. Later when the preferred tender has been selected the signer can be identified, whereas the signers of all other tenders will remain anonymous. If the signer regrets his tender, the identity of the signer can be computed without his cooperation.

Group signatures should not be confused with the related notion of group oriented signatures where certain subsets of a group of people are allowed to sign on behalf of the group. Such schemes do not provide a method for identifying the signers (see [D93]). Another related concept is that of multi-signatures which require a digital signature by many persons (see [Ok88] and [OO93]).

Four group signature schemes were presented by Chaum and Heijst in [CvH91] (also in [H92]). In their schemes, each group member U_i holds a secret key s_i. The signature signed using any of the s_i's can be verified using a public key K of the group.

In their first scheme, K is a list of public keys. The correspondence between the public keys and the signers is only known by a trusted authority. A signature is valid if and only if it is valid with respect to one of the public keys in the list K. The signer can be identified by the authority. The privacy of the signer is protected unconditionally. It is clear that if new group members join in, all old members have to change their keys. Otherwise, the recipient can distinguish the signatures of old members from those of new members.

In the other three schemes, K is a product of all or a part of the signers' public keys. The signatures are undeniable signatures (see [Ch91]). In order to identify the signer, each group member must execute an interactive protocol with the recipient to disavow the signature. If there is only one member who fails in executing the protocol, he must be the signer. But there is no reason to suppose all the group members are honest. If more members fail in executing the disavowal protocol, then the signer cannot be identified correctly. The anonymity in these three schemes depends on the difficulty of either factoring or computing discrete logarithms.

One of the open problems in [CvH91] is: is it possible to construct a scheme in which certain subset of group members can identify the signer?

In this chapter, group signature will be defined formally. A class of group signature schemes based on proofs of knowledge of one out of many witnesses are presented. The schemes can give unconditional anonymity and if more efficient schemes are required, anonymity can be obtained under some computational assumptions.

All schemes allow the group to be changed without the members having to select new keys. This property makes it possible for a member to sign on behalf of a subset of the group. By holding some auxiliary information for each signer, an authority can identify the signer without any cooperation of the group members.

Furthermore, the auxiliary information used to identify the signer can very easily be shared among the group members such that for some $n_1 \geq 1$, any set of n_1 group members can identify the signer. This solves the open

problem in [CvH91] mentioned before.

Finally the lower bounds on size of secret keys and auxiliary information are given. It is shown that in order to sign more messages for a signer with unconditional anonymity, the secret keys must become longer. Also entropy of the auxiliary information for identifying the signers depends on the number of signatures and the number of the group members linearly.

Part of this chapter is based on [ChePe94a].

6.2 Definitions

In this section, a definition of group signatures and their properties will be given. Suppose that all the signatures are defined on a message space $\mathcal{M}$.

Definition 33 (group signature) A group signature for a group of n members $U_1, \ldots, U_n$ and an authority A is a tuple $(k, gen, sign, test, iden)$. Here k is the security parameter, and $gen, sign, test, iden$ are all polynomial time (in k) algorithms.

- *gen* generates the keys. On input (k, n) and some random bits $\underline{b}$, it outputs

$$(pk, (s_1, s_2, \ldots, s_n, aux),$$

  where pk is the public key, s_i is the secret key for U_i, $i = 1, 2, \ldots, n$, and aux is auxiliary information for A.

- *sign* is a *probabilistic* algorithm to sign messages. On input s_i and $m \in \mathcal{M}$, it outputs $sign(s_i, m)$. A string σ is called a *correct signature* on message m, if there exists $i \in \{1, 2, \ldots, n\}$ such that $\sigma = sign(s_i, m)$.

- *test* is used to test signatures. On input pk, m, and a possible signature on m, it outputs *true* or *false*. A string σ is called an *acceptable signature* on m with respect to pk if $test(pk, m, \sigma) = true$.

- *iden* is used to identify the signer. On input m, an acceptable signature on m, and aux, it outputs $i \in \{1, 2, \ldots, n\} \cup \{?\}$.

For any $i \in \{1, 2, \ldots, n\}$, and any $m \in \mathcal{M}$, the scheme satisfies

$$test(pk, m, sign(s_i, m)) = true,$$

and

$$iden(aux, m, sign(s_i, m)) = i.$$

Remark From the definition above, it is clear that different secret keys produce different signatures, *i.e.* for any $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$, any message m, $sign(s_i, m) \neq sign(s_j, m)$.

Remark A correct signature on m is an acceptable signature on m. But an acceptable signature on m is not necessarily a correct signature on m.

Several properties of group signatures are defined in the following. First, it must be infeasible to forge signatures in adaptively chosen message attack (see [GMR88]).

Let $\mathcal{F}$ denote a polynomial time algorithm, which on input pk, and possibly aux, works as follows.

1. Repeat the following a polynomial number of times.
   (a) Generate a message $m \in \mathcal{M}$ and $i \in \{1, 2, \ldots, n\}$;
   (b) Get $sign(s_i, m)$.
2. Output a message $m_0 \in \mathcal{M}$ different from all m's in 1, and $\tilde{\sigma}(m_0)$.

Definition 34 (against forgery) Let (*k,gen,sign,test,iden*) be a group signature. The scheme is secure against forgery if the following holds: For any polynomial time $\mathcal{F}$ as above, for a large fraction of the keys,

$$\forall c > 0, \exists k_0 : \forall k > k_0$$
$$Prob[test(pk, m_0, \tilde{\sigma}(m_0)) = true] \leq k^{-c},$$

where $(m_0, \tilde{\sigma}(m_0))$ is the output of $\mathcal{F}$. The probability is over the random coins of signatures and the random coins of $\mathcal{F}$.

For group signature schemes, any group member should be able to sign the message on behalf of group without leaking his identity. Unconditional

anonymity is defined. It means that even with unlimited power, the signer can not be identified without *aux* as auxiliary information. With unlimited power, an adversary might be able to compute some possible secret keys. So it is necessary to consider the distribution of secret keys.

A public key pk produced by gen, corresponds to a set of possible secret keys defined as

$$SK(pk) = \{(sk_1, \ldots, sk_n) \mid \exists aux, gen(n,k) = (pk, (sk_1, \ldots, sk_n), aux)\},$$

where sk_i is a possible secret key of U_i, $i = 1, 2, \ldots, n$. We will omit pk in the following. The set $SK^{(i)}$ is defined as all the possible secret keys held by U_i, $i = 1, 2, \ldots, n$, *i.e.* $SK^{(i)}$ is the projection on i'th coordinate of SK. It is clear that

$$SK \subseteq SK^{(1)} \times SK^{(2)} \times \cdots \times SK^{(n)}.$$

If s_i denotes the secret key of U_i then

$$s_i \in SK^{(i)}, \quad i = 1, 2, \ldots, n$$

and

$$(s_1, s_2, \ldots, s_n) \in SK.$$

For any subset J of $\{1, 2, \ldots, n\}$, for any positive integers t and L, $0 < L \leq |J|t$, define a subset of J^L,

$$\mathcal{I}_J(t, L) = \{\underline{i} = (i_1, i_2, \ldots, i_L) \mid |\{j \mid i_j = i\}| \leq t, i_j \in J\}.$$

An intuitive explanation for $\mathcal{I}_J(t, L)$ is that if $\underline{i} = (i_1, i_2, \ldots, i_L) \in \mathcal{I}_J(t, L)$, any element of J appears in $(i_1, i_2, \ldots, i_L)$ not more than t times. When $J = (i_1, i_2, \ldots, i_L)$, $\mathcal{I}_J(t, L)$ will be simply denoted as $\mathcal{I}(t, L)$.

For any correct signature $\sigma(m)$ on message m, denote the event that "there is a possible secret key sk for U_r such that $\sigma(m) = sign(sk, m)$" as "$\sigma(m) \Leftarrow r$". It means that $\sigma(m)$ is possibly made by U_r.

For any L messages

$$\underline{m} = (m_1, m_2, \ldots, m_L)$$

and L correct signatures

$$\sigma(\underline{m}) = \sigma(m_1), \sigma(m_2), \ldots, \sigma(m_L),$$

and for any $\underline{i} = (i_1, i_2, \ldots, i_L) \in \mathcal{I}_J(T, L)$, "$\sigma(\underline{m}) \Leftarrow \underline{i}$" denotes the event that $\sigma(m_j)$ is made by U_{i_j}, $j = 1, 2, \ldots, L$. Note that if $\sigma(m_j)$ and $\sigma(m_{j'})$ are both made by U_r, then P_r has to use the same secret key to sign both m_j and $m_{j'}$. So precisely, "$\sigma(\underline{m}) \Leftarrow \underline{i}$" means that there exists $sk \in SK$ such that for all $j \in \{1, 2, \ldots, L\}$,

$$sign(sk_{i_j}, m_{i_j}) = \sigma(m_{i_j}),$$

where sk_{i_j} is the projection of sk on i_j-th coordinate.

Intuitively, unconditional anonymity requires that any correct signature $\sigma(m)$ could have been made by any of group members equally likely. This fact can be formulated as

$$prob[\sigma(m) \Leftarrow r] = \frac{1}{n}, \quad r = 1, 2, \ldots, n.$$

A more general definition of anonymity is given as follows.

Definition 35 (anonymity) Let (*k,gen,sign,test,iden*) be a group signature. The scheme provides anonymity for signing T messages if for any $J \subseteq \{1, 2, \ldots, n\}$, for any $L \leq |J|T$, the following holds: For any L different messages

$$\underline{m} = (m_1, m_2, \ldots, m_L),$$

given L correct signatures from $\{P_j\}_{j \in J}$

$$\sigma(\underline{m}) = (\sigma(m_1), \sigma(m_2), \ldots, \sigma(m_L))$$

such that each of the members in $\{P_j\}_{j \in J}$ has made not more than T signatures, then for any $\underline{i} = (i_1, i_2, \ldots, i_L) \in \mathcal{I}_J(T, L)$,

$$Prob[\sigma(\underline{m} \Leftarrow \underline{i}] = \frac{1}{|\mathcal{I}_J(T, L)|}.$$

The probability is over the choice of $sk \in SK$ and the random coins used in signature.

Remark Under the definition, each group member can sign at least T messages without losing the unconditional anonymity. It can be generalized to let each P_i sign at least T_i messages, $i = 1, 2, \ldots, n$.

Group signatures also require the signer to bear responsibility for the signatures he has made. The authority can identify the signer from the signature with *aux* as auxiliary information.

For any subset J of $\{1, 2, \ldots, n\}$, let $\mathcal{F}_J$ denote a polynomial time algorithm, which on input pk, $\{s_j\}_{j \in J}$, works as follows:

1. Repeat the following a polynomial number of times.
   (a) Generate a message $m \in \mathcal{M}$, and a number $i \in J^c$;
   (b) Get $sign(s_i, m)$.
2. Output a message $m' \in \mathcal{M}$ different from all m's in 1 and an acceptable signature $\sigma(m')$ on m'.

Definition 36 (signer identification) Let (*k,gen,sign,test,iden*) be a group signature. The scheme provides signer identification if the following holds: For any subset J of $\{1, 2, \ldots, n\}$, and for any polynomial time algorithm $\mathcal{F}_J$ as above,

$$\forall d > 0, \exists k_0 : \ \forall k > k_0$$
$$Prob[iden(aux, m', \sigma(m')) \in J] \geq 1 - k^{-d},$$

where $(m', \sigma(m'))$ is the output of $\mathcal{F}_J$. The probability is over the random coins of $\mathcal{F}_J$.

If $|J| = 1$, then this definition means that the signer must be identified by the authority with overwhelming probability.

Definition 37 (secure group signature) A group signature scheme is secure, if it is secure against forgery, provides anonymity for signing T messages, where T is a positive integer, and signer identification.

6.3 Proof of one out of n witnesses

All group signature schemes proposed in this chapter are based on the proofs of one out of many witnesses given by Schoenmakers in [Sm93] and presented in [CDS94].

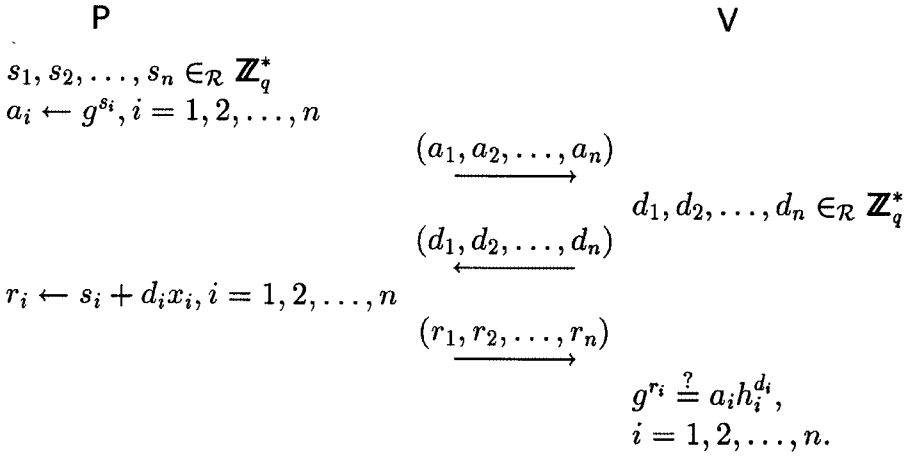

Figure 6.1: Proving knowledge of n witnesses

Let G_q denote a group of prime order q and let g be a generator of G_q. The common input to the prover and verifier is $(g, h_1, \ldots, h_n)$ for some $n \in \mathbb{N}$, where each $h_i \in G_q$. Let $h_i = g^{x_i}$, $i = 1, 2, \ldots, n$. Suppose the prover knows all x_i's. The protocol shown in Figure 6.1 is a proof of knowledge of x_i, $i = 1, 2, \ldots, n$.

Now suppose that the prover only knows one of the n witnesses. Given one of x_i's as secret input, the prover shows that he knows w such that for some $i \in \{1, 2, \ldots, n\}$: $h_i = g^w$. The protocol is sketched in Figure 6.2 for the case $w = x_1$.

The idea behind this protocol is that the challenge $c = \sum_1^n d_i$, allows the prover to choose $(n-1)$ but not all of the d_j's beforehand. In order to reply all d_j's, the prover must know at least one of n witnesses. However the proof does not reveal which witness the prover knows since no information about which d_j's prover chooses is contained in the messages from prover. The formal result and proof about the protocol were given in [Sm93].

Theorem 38 *The protocol in Figure 6.2 is a witness indistinguishable proof of knowledge (see [FS90]) of w satisfying*

$$h_i = g^w \quad \textit{for some } i \in \{1, 2, \ldots, n\}.$$

Remark An extension of this protocol allows the prover to show that he knows at least k out of n secret keys (see [Sm93]).

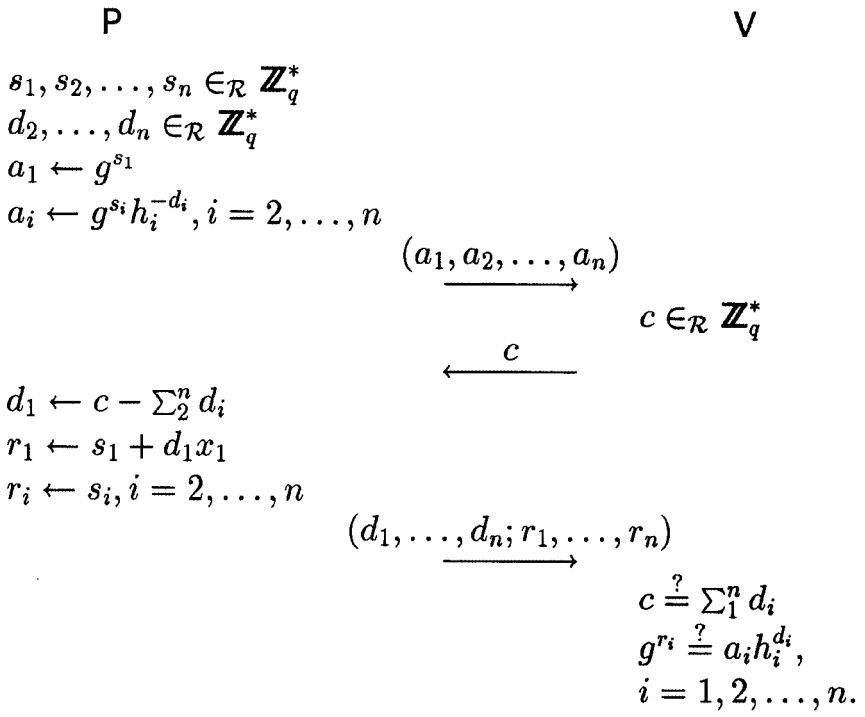

Figure 6.2: Proving knowledge of one of n witnesses

6.4 Scheme with unconditional anonymity

This section presents a group signature scheme giving unconditional anonymity. We only consider the case with two persons (P_1 and P_2) in the group (the general case is obtained by a straightforward extension).

6.4.1 Signing one message

Let two generators g_1 and g_2 of G_q be given (the actual selection of these generators is not important as long as no group member can express one as the power of the other). The secret key of P_i is $(x_{i1}, x_{i2}) \in \mathbb{Z}_q^2$ for $i = 1, 2$. The public key of the group is (g_1, g_2, h_1, h_2) where $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$ for $i = 1, 2$ (assume $h_1 \neq h_2$). Note that it is known that h_i is the public key of P_i.

P_i's signature on a message $m = (m_1, m_2)$ is $z = m_1^{x_{i1}} m_2^{x_{i2}}$ plus a proof that this is correct with respect to either h_1 or h_2. This proof will be shown

in Figure 6.3 and then its application to the group signature is described.

Using the arguments in [Sm93] it can be shown that the protocol in Figure 6.3 constitutes a proof that the prover knows a pair (s,t) such that

$$z = m_1^s m_2^t \quad \wedge \quad (h_1 = g_1^s g_2^t \quad \vee \quad h_2 = g_1^s g_2^t).$$

If $\log_{m_1} m_2 \neq \log_{g_1} g_2$, there are two possible witnesses to this claim: a pair (s,t) satisfying $h_1 = g_1^s g_2^t$ and a pair (s',t') satisfying $h_2 = g_1^{s'} g_2^{t'}$.

Lemma 39 *If $\log_{m_1} m_2 \neq \log_{g_1} g_2$ the proof is witness indistinguisable (see [FS90]).*

Proof First it will proved that given z there is exactly one pair (α_1, α_2) such that

$$h_1 = g_1^{\alpha_1} g_2^{\alpha_2} \text{ and } z = m_1^{\alpha_1} m_2^{\alpha_2}.$$

Suppose that $g_2 = g_1^{\delta_1}$ and $m_2 = m_1^{\delta_2}$, also $h_1 = g_1^{\varphi_1}$ and $z = m_1^{\varphi_2}$. Then since $\delta_1 \neq \delta_2$, the group of equations

$$\begin{cases} \alpha_1 + \alpha_2 \delta_1 = \varphi_1 \\ \alpha_1 + \alpha_2 \delta_2 = \varphi_2 \end{cases}$$

has one and only one solution (α_1, α_2).

Similarly, there is exactly one pair (β_1, β_2) such that

$$z = m_1^{\beta_1} m_2^{\beta_2} \text{ and } h_2 = g_1^{\beta_1} g_2^{\beta_2}.$$

We have to show that a prover, knowing (α_1, α_2) would construct messages with the same distribution as a prover knowing (β_1, β_2) (the protocol for a prover knowing the witness to h_2 is symmetric).

For the same reason as above, given a_1, a_2, b_1, b_2 there is exactly one possible tuple (v_1, v_2, w_I, w_2) such that

$$a_1 = g_1^{v_1} g_2^{v_2}, \quad b_1 = m_1^{v_1} m_2^{m_2}, \quad a_2 = g_1^{w_1} g_2^{w_2} \text{ and } b_2 = m_1^{w_1} m_2^{w_2}.$$

These numbers contain no Shannon information about which of d_j's is chosen beforehand. Furthermore, the numbers (r_1, r_2, u_1, u_2) are uniquely determined by:

$$r_1 = v_1 + d_1 \alpha_1, \quad r_2 = v_2 + d_1 \alpha_2, \quad u_1 = w_1 + d_2 \beta_1 \text{ and } u_2 = w_2 + d_2 \beta_2.$$

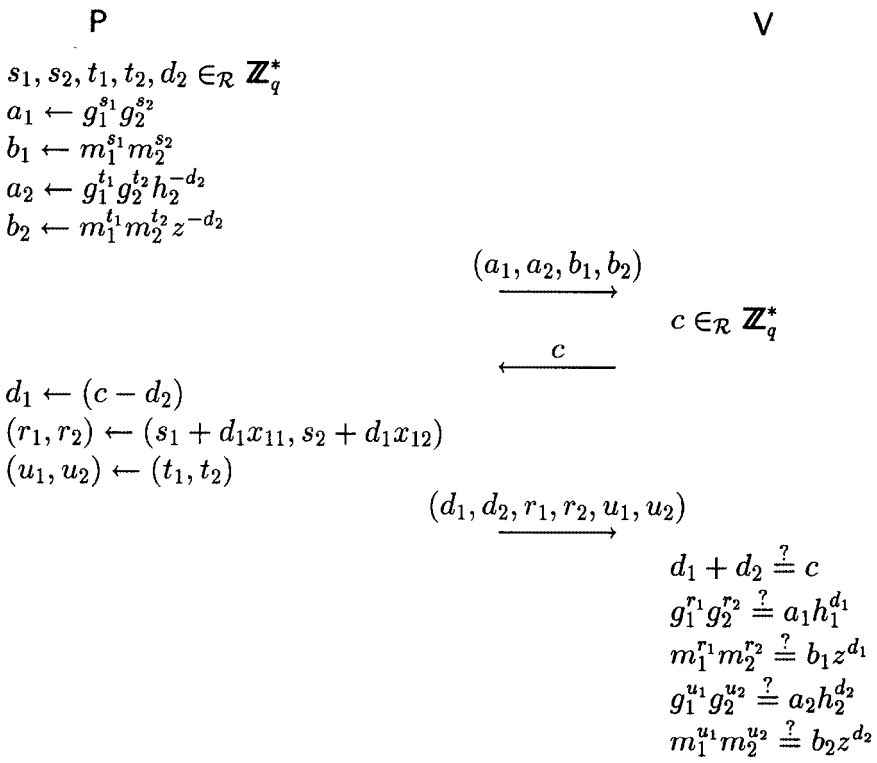

Figure 6.3: Proving that z is constructed correctly with respect to h_1

Thus the messages sent by the prover reveal no information about which of the two witnesses he knows. □

Given three hash-functions,[1] $\mathcal{H}$, $\mathcal{H}_1$ and $\mathcal{H}_2$, P_i now signs a message m as follows:

1. Compute from m a pair $(m_1, m_2) \in G_q^2$ as $m_j = \mathcal{H}_j(m)$ for $j = 1, 2$.
2. P_i computes z and executes the proof, computing

$$c = \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2).$$

[1]By choosing two random, but fixed strings ρ_1 and ρ_2, $\mathcal{H}_1(m)$ and $\mathcal{H}_2(m)$ can for example be defined as $\mathcal{H}(\rho_j, m)$ for $j = 1$ and $j = 2$, respectively. Thus only one hash function is really needed.

Here $\mathcal{H}$ is supposed to be "pseudo-random" as required by Assumption 1 in Chapter 4.

3. The signature on m is $(z, d_1, d_2, r_1, r_2, u_1, u_2)$. It is verified that

$$d_1 + d_2 = \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2),$$

where $a_1 = g_1^{r_1} g_2^{r_2} h_1^{-d_1}$, $b_1 = m_1^{r_1} m_2^{r_2} x^{-d_1}$, $a_2 = g_1^{u_1} g_2^{u_2} h_2^{-d_2}$, $b_2 = m_1^{u_1} m_2^{u_2} z^{-d_2}$.

Proposition 40 *The group signature scheme presented above satisfies:*

1. *Given a correct gignature $\sigma(m)$, if $m = (m_1, m_2)$ and $\log_{g_1} g_2 \neq \log_{m_1} m_2$, then for any $r \in \{1, 2\}$,*

$$Prob[\sigma(m) \Leftarrow r] = \frac{1}{2}.$$

2. *Given two correct signatures $\sigma(m)$ and $\sigma(m')$, if $m = (m_1, m_2)$, $\log_{g_1} g_2 \neq \log_{m_1} m_2$, and $m' = (m'_1, m'_2)$, $\log_{g_1} g_2 \neq \log_{m'_1} m'_2$, if each of P_1 and P_2 signs one message, then for any tuple $\underline{i} \in \{(1, 2), (2, 1)\}$,*

$$Prob[\sigma(m, m') \Leftarrow \underline{i}] = \frac{1}{2}.$$

Proof For one correct signature $\sigma(m)$, where $m = (m_1, m_2)$, by the proof of Lemma 39, if $\log_{g_1} g_2 \neq \log_{m_1} m_2$, then there exists exactly one $sk_1 = (\alpha_1, \alpha_2)$ which is a possible secret key of P_1 such that

$$\sigma(m) = sign(sk_1, m).$$

Similarly, there exists exactly one $sk_2 = (\beta_1, \beta_2)$ which is a possible secret key of P_2 such that

$$\sigma(m) = sign(sk_2, m).$$

So $\sigma(m)$ is made by sk_1 and sk_2 equally likely.

For two correct signatures $\sigma(m)$ and $\sigma(m')$, where $m = (m_1, m_2)$ and $m' = (m'_1, m'_2)$, if $\log_{g_1} g_2 \neq \log_{m_1} m_2$ and $\log_{g_1} g_2 \neq \log_{m'_1} m'_2$, then by the

same argument as in the proof of Lemma 39, there exist sk_1 and sk_2, possible secret keys of P_1 and P_2 separately, such that

$$\sigma(m) = sign(sk_1, m) = sign(sk_2, m).$$

Similarly, there exist sk_1' and sk_2', possible secret keys of P_1 and P_2 separately, satisfying

$$\sigma(m') = sign(sk_1', m') = sign(sk_2', m').$$

If each of P_1 and P_2 signs one message, then with very large probability,

$$sk_1 \neq sk_1' \quad \text{and} \quad sk_2 \neq sk_2'$$

Then

$$\text{“}\sigma(m, m') \Leftarrow (1, 2)'' \quad \text{and} \quad \text{“}\sigma(m, m') \Leftarrow (2, 1)''$$

are equally likely. □

By the definition of anonymity, with a restriction on message space, Proposition 40 is equivalent to the following theorem.

Theorem 41 *The group signature scheme described above provides anonymity for signing* 1 *message, if for all* $m = (m_1, m_2) \in \mathcal{M}$, $\log_{g_1} g_2 \neq \log_{m_1} m_2$.

Remark If m is chosen randomly, where $m = (m_1, m_2)$ is computed by $m_i = \mathcal{H}(\rho_i, m)$, $i = 1, 2$, and if $\mathcal{H}$ works as a random oracle, then with probability $1 - \frac{1}{q}$,

$$\log_{g_1} g_2 \neq \log_{m_1} m_2.$$

If P_i signs two different messages, then an unlimited powerful receiver can easily tell that both signatures correspond to h_i. In fact, if the notation in the proof of Proposition 40 is used in this case, then either $sk_1 = sk_1'$ or $sk_2 = sk_2'$, from which it can be judged who has made two signatures.

The security of the scheme against forgery depends on the property of the hash function. The investigation of this is exactly the same as for general Fiat-Shamir style signature schemes based on three move witness hiding proofs of knowledge (see [FS87]).

Before a member signs a message he is unconditionally protected against framing, since in each $SK^{(i)}$ there are q pairs of (s,t) such that $h_i = g_1^s g_2^t$ which are all possible secret keys of P_i. So the probability of being framed is very small. But after he signs a message m, if someone can find out $\sigma(m)$ is his signature, then he can be framed (given sufficient computing power), since it can be determined uniquely that $s_i = (\alpha_1, \alpha_2)$, satisfying $h_i = g_1^{\alpha_1} g_2^{\alpha_2}$ and $z = m_1^{\alpha_1} m_2^{\alpha_2}$ is the secret key of P_i.

Finally, we note that this scheme provides the anonymity of each P_i even if it is known that h_i is the public key of P_i. This implies that the members do not have to select a new pair of keys when the group is changed (e.g., when new members join the group).

6.4.2 Signing T messages

The above scheme can in many ways be extended to signing T messages. The following sketches one possibility. Let $T+1$ generators $g_1, \ldots, g_{T+1}$ of G_q be given. The secret key of P_i is $(x_{i1}, \ldots, x_{i,T+1}) \in \mathbb{Z}_q^{T+1}$ for $i = 1, 2$. The public key of the group is

$$(g_1, \ldots, g_{T+1}, h_1, h_2)$$

where $h_i = g_1^{x_{i1}} \cdots g_{T+1}^{x_{i,T+1}}$ for $i = 1, 2$ (assume $h_1 \neq h_2$).

P_i's signature on a message $m = (m_1, \ldots, m_{T+1})$ is $z = m_1^{x_{i1}} \cdots, m_{T+1}^{x_{i,T+1}}$ plus a proof that this is correct with respect to either h_1 or h_2. A witness indistinguishable proof of this can be constructed by modifying the protocol in Figure 6.3. The digital signature is then obtained as before.

(T,l)-Condition In the scheme for signing T messages, for any integer l, $l \leq T$, l messages $m_t = (m_{t,1}, m_{t,2}, \ldots, m_{t,T+1})$, $t = 1, 2, \ldots, l$ satisfy (T,l)-condition, if for $g_j = g_1^{\delta_j}$, and $m_{t,j} = m_{t,1}^{\eta_{t,j}}$, $j = 2, \ldots, T+1$, the following matrix has rank $l+1$,

$$\begin{pmatrix} 1 & \delta_2 & \cdots & \delta_{T+1} \\ 1 & \eta_{1,2} & \cdots & \eta_{1,T+1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \eta_{l,2} & \cdots & \eta_{l,T+1} \end{pmatrix}$$

Remark (T,l)-condition generalizes the condition that $\log_{g_1} g_2 \neq \log_{m_1} m_2$ in Lemma 39.

Proposition 42 *In the scheme for signing T messages, for any l correct signutures $\sigma(m_1), \sigma(m_2), \ldots, \sigma(m_l)$, $0 \leq l \leq T$, if the messages $m_1, m_2, \ldots, m_l$ satisfy (T, l)-conditions, then for any group member P_i, there are q^{T-l} possible $(T+1)$-tupies in $\mathbb{Z}_q$, which could be his secret key.*

Proof For any P_i suppose that

$$(\alpha_1, \alpha_2, \ldots, \alpha_{T+1})$$

is his secret key. Then

$$g_1^{\alpha_1} g_2^{\alpha_2} \cdots g_{T+1}^{\alpha_{T+1}} = h_i.$$

If $h_i = g_1^{\varphi_0^{(i)}}$, and $g_j = g_1^{\delta_j}, j = 2, \ldots, T+1$, then

$$\alpha_1 + \alpha_2\delta_2 + \cdots + \alpha_{T+1}\delta_{T+1} = \varphi_0^{(i)} \qquad (1).$$

If the signatures are

$$\sigma(m_t) = (z_t, d_{t1}, d_{t2}, r_{t1}, r_{t2}, u_{t1}, u_{t2}),$$

where $m_t = (m_{t,1}, \ldots, m_{t,T+1})$ and if $m_{t,j} = m_{t,1}^{\eta_{t,j}}$, $j = 2, \ldots, T+1$, $z_t = m_{t,1}^{\varphi_t}$, then from

$$m_{t,1}^{\alpha_1} m_{t,2}^{\alpha_2} \cdots m_{t,T+1}^{\alpha_{T+1}} = z_t$$

l equations

$$\alpha_1 + \alpha_2\eta_{t2} + \cdots + \alpha_{T+1}\eta_{t,T+1} = \varphi_t \qquad (t),$$

$t = 1, 2, \ldots, l$, can be obtained.

This gives $l+1$ equations for the secret key $(\alpha_1, \alpha_2, \ldots, \alpha_{T+1})$. Since messages satisfy (T, l)-condition, there exist q^{T-l} solutions, which are all possible secret keys, since all the proofs are witness indistinguishable. □

In order to get unconditional anonymity, a restriction about messages must be considered. It can be shown that if l messages are chosen randomly then with quite large probability, the messages will satisfy (T, l)-condition.

Theorem 43 *The group signature acheme described above provides anonymity for signing T messages, if any of T different messages satisfy (T, T)-condition.*

Proof If any T different messages satisfy (T, T)-condition, then for any $l \leq T$, any l different messages satisfy (T, l)-condition. By Proposition 42, for any l correct signatures on l different messages, it is made by any of members equally likely. If L correct signatures on L different messages are made by a subset of group members such that each of them makes not more than T signatures, then L signatures contain no information about which member has made which signatures. □

Remark The group signature scheme can be improved such that anonymity does not depend on (T, T) condition about messages (see [ChePe94b]).

6.4.3 Identifying the signer

This section describes a general way of identifying the signer of group signatures. Let the group members be $P_1, P_2, \ldots, P_n$ for some $n \in \mathbb{N}$, and let P_i hold a secret key, s_i. The public key of the group is denoted by K, and an acceptable signature on the message, m, with respect to this public key is denoted by $\sigma_K(m)$. Suppose that a given scheme satisfies all requirements to group signatures except that the signer cannot be identified. In order to add this, the group selects two public keys and each member gets two secret keys in the given scheme. Let the secret key of P_i be (s_i, t_i) and the public key of the group be (K_1, K_2). Using s_i (t_i), P_i makes signatures with respect to K_1 (K_2).

A pair (σ_1, σ_2) is a valid signature on m with respect to (K_1, K_2) if

$$\sigma_1 = \sigma_{K_1} \quad \text{and} \quad \sigma_2 = \sigma_{K_2}$$

Thus each member signs a message by signing it twice, first using s_i and then using t_i. The authority is given $(t_1, \ldots, t_n)$ as auxiliary information (and the identity of the member having t_i as secret key). Since by the definition, different secret keys produce different signatures, *i.e.* for any $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$, any message m, $sign(t_i, m) \neq sign(t_j, m)$, this information enables the authority to identify the signer from σ_{K_2}, but of course not to sign (unless it can forge signatures with respect to K_1).

This way of identifying signer will be called double-signing. Using double-signing the signer in three of the schemes in [CvH91] can be identified much easier than using the interactive protocols proposed there (at the cost of twice as long signatures).

Let us see how double-signing works for the schemes proposed in this section. For the sake of simplicity, only the scheme for signing one message will be considered. Here, P_i holds the secret key (x_{i1}, x_{i2}) corresponding to public key $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$, $i = 1, 2$. For any signature

$$\sigma(m) = (z, d_1, d_2, u_1, u_2, r_1, r_2),$$

as shown in the proof of Lemma 39, there exists exactly one pair (α_1, α_2) such that

$$h_1 = g_1^{\alpha_1} g_2^{\alpha_2} \quad \text{and} \quad z = m_1^{\alpha_1} m_2^{\alpha_2}$$

Also there is exactly one pair (β_1, β_2) such that

$$h_2 = g_1^{\beta_1} g_2^{\beta_2} \quad \text{and} \quad z = m_1^{\beta_1} m_2^{\beta_2}$$

Notice that if P_1 is the signer then $(\alpha_1, \alpha_2) = (x_{11}, x_{12})$, whereas with very high probability $(\beta_1, \beta_2) \neq (x_{21}, x_{22})$. Thus, a person knowing the secret keys of P_1 and P_2 can tell whether P_1 or P_2 is the signer. Therefore, double-signing can be used to turn this scheme into a group signature in which the authority can easily determine the signer.

When using double-signing each member has two secret keys s_i and t_i, one of which, say t_i is sent to the authority. If this key is shared varifiably among the n group members in a n_1 out of n threshold scheme any n_1 members can identify the signer.

For the schemes presented here the distribution as well as the identification can be done quite efficiently. The main idea will be sketched for the scheme in Section 6.4.1.

Suppose that the secret key for P_i is $((s_{i1}, s_{i2}), (t_{i1}, t_{i2})$ and the public key is (h_i, k_i), where $h_i = g_1^{s_{i1}} g_2^{s_{i2}}$ and $k_i = g_1^{t_{i1}} g_2^{t_{i2}}$. Using the non-interactive, verifiable secret sharing scheme from [P92]), each P_i shares his secret key (t_{i1}, t_{i2}) verifiably among all n group members. This can be done quite efficiently using the fact that k_i is known. Each member P_j gets a share (t_{ij1}, t_{ij2}) of this key, for which $g_1^{t_{ij1}} g_2^{t_{ij2}}$ is publicly known (see [P91]).

Later, when n_1 members want to decide whether a given signature z on a message m was made by P_i, each computes $m_1^{t_{ij1}} m_2^{t_{ij2}}$. These partial results can then easily be combined into $m_1^{t_{i1}} m_2^{t_{i2}}$. They conclude that P_i was indeed the signer if and only if this equals z.

Remark A member able to compute discrete logarithms can make a group signature for which no other member will be held responsible.

6.5 Lower bound on the secret keys

It might be noticed in Section 6.4.2 that in order to sign more messages anonymously, the secret key for each signer becomes longer. In this section, it will be shown that this cannot be avoided. The main idea for proving the lower bound of the secret keys is to divide the set of possible secret keys of each member to some nonempty subsets. First, it will be proved that each subset contains at least one secret key. Then by proving that the intersection of any two different subsets is empty, the number of possible secret keys must be larger than or equal to the number of subsets.

For a t-tuple $\underline{i} = (i_1, i_2, \ldots, i_t) \in \{1, 2, \ldots, n\}^t$, and t different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$, for every $1 \leq r \leq n$, the following set is defined:

$$SK_{\underline{i}}^{(r)}(\underline{m}) = \{sk \in SK^{(r)} \mid sign(sk, m_j) = sign(s_{i_j}, m_j), j = 1, 2, \ldots, t\}$$

where s_i is the secret key of P_i $(i = 1, 2, \ldots, n)$.

$SK_{\underline{i}}^{(r)}(\underline{m})$ is the set of all the possible keys of P_r which can be used to sign all $\underline{m} = (m_1, m_2, \ldots, m_t)$ such that the signatures are the same as P_{i_j} signs m_j, $j = 1, 2, \ldots, t$.

Consider a simple situation: $t = 1$. If for an i in $\{1, 2, \ldots, n\}$ and a message m, for some r, $SK_i^{(r)}(m) = \emptyset$, then it can be excluded that $\sigma(m) = sign(s_i, m)$ is signed by P_r, which contradicts anonymity. The following lemma generalizes this.

Lemma 44 *If a group signature (k,gen,sign,test,iden) provides anonymity for signing T messages, then for any $t \leq T$, the following holds: For all $\underline{i} = (i_1, i_2, \ldots, i_t)$, and any t different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$,*

$$SK_{\underline{i}}^{(r)}(\underline{m}) \neq \emptyset$$

$r = 1, 2, \ldots, n$.

Proof If there exist $t \leq T$ different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$, and a t-tuple $\underline{i} = (i_1, i_2, \ldots, i_t)$, such that

$$SK_{\underline{i}}^{(r_0)}(\underline{m}) \neq \emptyset$$

for some r_0, then for $\sigma(m_j) = sign(s_{i_j}, m_j)$, $j = 1, 2, \ldots, t$, if t-tuple $\underline{i}_0 = (r_0, r_0, \ldots, r_0)$

$$Prob[\sigma(\underline{m}) \Leftarrow \underline{i}_0] = 0,$$

which contradicts the definition of anonymity. □

Theorem 45 *Let a group signature (k,gen,sign,test,iden) be given. If it provides anonymity for signing T messages, then for any $r \in \{1, 2, \ldots, n\}$,*

$$[SK^{(r)}| \geq n^T.$$

Proof First, for any $t \leq T$ different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$, if

$$\underline{i} = (i_1, i_2, \ldots, i_t) \text{ and } i' = (i'_1, i'_2, \ldots, i'_t),$$

$\underline{i} \neq \underline{i'}$, then

$$SK_{\underline{i}}^{(r)}(\underline{m}) \cap SK_{\underline{i'}}^{(r)}(\underline{m}) = \emptyset$$

Otherwise there exists

$$sk \in SK_{\underline{i}}^{(r)}(\underline{m}) \cap SK_{\underline{i'}}^{(r)}(\underline{m})$$

such that for some j, $j \in \{1, 2, \ldots, n\}$, $i_j \neq i'_j$,

$$sign(sk, m_j) = sign(s_{i_j}, m_j) \text{ and } sign(sk, m_j) = sign(s_{i'_j}, m_j),$$

which is impossible by the definition of group signatures, since different members must make different signatures.

Second, by Lemma 44, for any t different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$, and any t-tuple $\underline{i} = (i_1, i_2, \ldots, i_t) \in \{1, 2, \ldots, n\}^t$,

$$|SK_{\underline{i}}^{(r)}(\underline{m})| \geq 1$$

Finally, for any t different messages $\underline{m} = (m_1, m_2, \ldots, m_t)$

$$|SK^{(r)}| \geq \sum_{\underline{i} \in \{1,2,\ldots,n\}^t} |SK_{\underline{i}}^{(r)}(\underline{m})| \geq n^t,$$

for any $t \leq T$. So

$$SK^{(r)} \geq n^T.$$

□

For any public key pk, there must be at least n^T possible secret keys in order to get unconditional anonymity for signing T messages. In other words, it takes at least $T \log n$ bits to represent the secret key of each group member. The size of the secret key grows linearly in T.

6.6 Length of the auxiliary information

In this section, we consider the length of the auxiliary information held by the authority in group signature schemes. First some random variables are defined for this purpose.

Definition 46 ((L,T)-history) For any $L, 0 < L \leq nT$, a tuple

$$hist_L(\underline{m}) = ((m_1, \sigma(m_1)), (m_2, \sigma(m_2)), \ldots, (m_L, \sigma(m_L)))$$

is called an (L,T)-history, if

$$\underline{m} = (m_1, m_2, \ldots, m_L),$$

are L different messages and there exists a tuple

$$\underline{i} = (i_1, i_2, \ldots, i_L) \in \mathcal{I}(T, L)$$

such that

$$\sigma(m_l) = sign(s_{i_l}, m_l), \quad l = 1, 2, \ldots, L.$$

Definition 47 (random variables) Let (*k,gen,sign,test,iden*), an integer L, $0 < L \leq nT$, and an (L,T)-history $hist_L(\underline{m})$ be given. *gen* defines a probability space. The following random variables are defined on it.

- A is the random variable of the auxiliary information.
- ID is the random variable defined as follows:

  $$ID = (i_1, i_2, \ldots, i_L) \in \mathcal{I}(T, L),$$

  if there exists $sk \in SK$ such that for all $l \in \{1, 2, \ldots, L\}$, $i_l = r$,

  $$\sigma(m_l) = sign(sk_r, m_l),$$

  where sk_r is the projection of sk on r-th coordinate, *i.e.*

  $$\sigma(\underline{m}) \Leftarrow \underline{i},$$

  where $\underline{i} = (i_1, i_2, \ldots, i_L)$.

From the definition of unconditional anonymity, the following lemma can be reduced directly.

Lemma 48 *If the group signature scheme (k,gen,sign,test,iden) provides anonymity for signing T messages, then for any (L,T)-history $hist_L(\underline{m})$, random variable ID defined as above has uniform distribution on $\mathcal{I}(T,L)$. Especially, the entropy of ID*

$$H(ID) = \log_2 |\mathcal{I}(T, L)|.$$

Theorem 49 *If the group signature acheme (k,gen,sign,test,iden) provides anonymity for signing T messages and signer identification, then*

$$H(A) \geq Tn(\log n - 1).$$

Proof The key point for proving the theorem is that by knowing the value of A, ID is determined uniquely since the scheme provides signer identification.

Let $L = Tn$, and consider an (L,T)-history, $hist_L(\underline{m})$, and random variable ID. The mutual entropy of A and ID can be written as

$$I(A, ID) = H(A) - H(A|ID) = H(ID) - H(ID|A).$$

Since it provides signer identification,

$$H(ID|A) = 0.$$

Then

$$H(A) = H(ID) + H(A|ID) \geq H(ID).$$

With the lemma above,

$$H(ID) = \log \frac{(Tn)!}{(T!)^n}.$$

Stirlings Formula,

$$n! \approx e^{-n} n^n \sqrt{2\pi n}$$

gives

$$\log \frac{(Tn)!}{(T!)^n} \approx Tn \log n + \log \sqrt{2\pi Tn} - n \log \sqrt{2\pi n} \geq Tn(\log n - 1).$$

So

$$H(A) \geq Tn(\log n - 1).$$

□

The information contained in A must increase as T or n increases in order to guarantee anonymity for signing T message and signer identification for a group with n members.

6.7 Scheme with computational anonymity

It has been proved that the length of the secret key unavoidably grows as the signer signs more messages with unconditional anonymity. In this section, more efficient group signature schemes will be proposed but with computational anonymity.

As before, let G_q be a group of prime order and let g be a generator of this group. Again the scheme will be described for groups consisting of two persons, P_1 and P_2.

The public key of the group is (g, h_1, h_2) and the secret key of P_i is $x_i = \log_g h_i$ for $i = 1, 2$. When signing a message $m \in G_q$, P_i computes $z_i = m^{x_i}$, chooses $z_{3-i} \in G_q$ at random and proves that he knows w such that

$$(h_1 = g^w \ \wedge \ z_1 = m^w) \quad \vee \quad (h_2 = g^w \ \vee \ z_2 = m^w) \qquad (*)$$

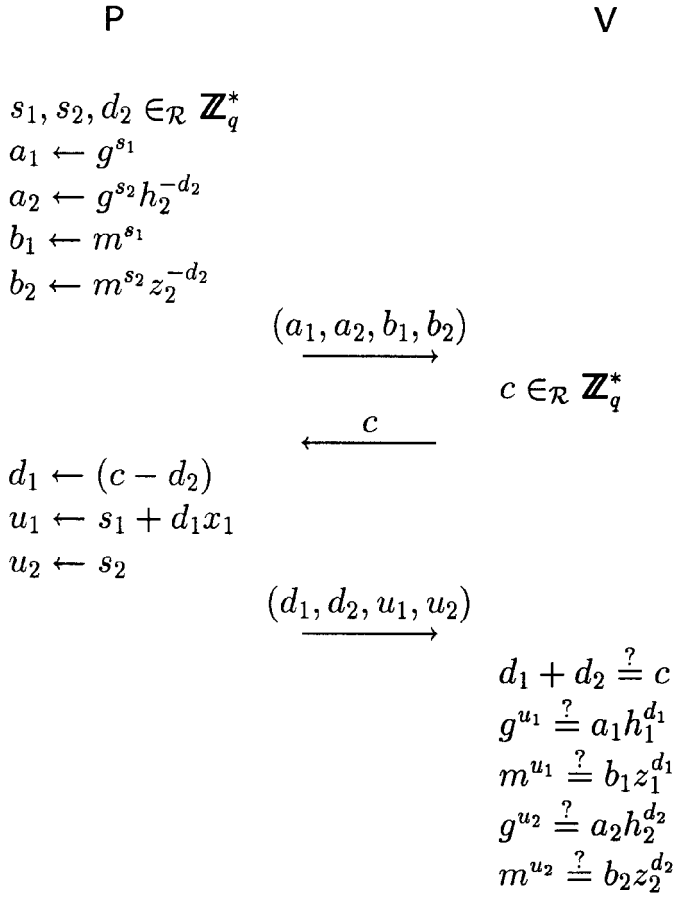

Figure 6.4: Proving $z_1 = m^{\log_g h_1}$ or $z_2 = m^{\log_g h_2}$

The common input is $(g, h_1, h_2, m, z_1, z_2)$ and the secret input of the prover is x_1 or x_2 (in Figure 6.4 the prover knows x_1 — the case of x_2 is symmetric). By a straightforward modification of [Sm93], it can be shown that the protocol is a proof of knowledge of a witness to $(*)$.

This protocol can be turned into a signature scheme as in Section 6.4.1 (and [FS87]). Next will be argued that the receiver of such signatures cannot tell whether the signature was made using x_1 or x_2. First note however, that the protocol is not witness indistinguishable in the sense of [FS90], where it is required that even a distinguisher who knows the possible witnesses cannot tell which witness the prover knows. That clearly does not hold for this protocol. Therefore the following contains a less formal argument for the anonymity of the group members. First, it is shown (based on a discrete logarithm assumption stated below) that if no group member has previously made any signatures it is infeasible to tell who made a given signature. Then it is argued that knowledge of other signatures plus the identity of the corresponding signer does not help the receiver deciding which member made a given signature. The following assumption underlies the security of the scheme:

Assumption 2 Let D be a probabilistic polynomial time machine which takes (g, h, m, z) as input. Let $Pr_=$ denote the probability that D outputs 1, when m is chosen uniformly at random and $\log_g h = \log_m z$. Let $Pr_{\neq}$ denote the probability that D outputs 1 when m and z are chosen uniformly at random. Then for all $D : |Pr_= - Pr_{\neq}|$ is negligible as a function of the order of the group. The probability is over the choice of m and random coins of D.

Now consider three possible provers:

P_0 : The input satisfies $z_1 = m^{x_1}$ and $z_2 = m^{x_2}$;
P_0 just chooses d_1 at random.

P_1 : The input satisfies $z_1 = m^{x_1}$ and z_2 is chosen uniformly at random;

P_2 : z_1 is chosen uniformly at random and $z_2 = m^{x_2}$.

Lemma 50 *Under Assumption 2 the following holds. No polynomially bounded verifier can distinguish between P_0 and P_2. Similarly, P_0 and P_1 cannot be distinguished.*

Proof Let a verifier $\tilde{V}$ be given.
Given g, h_1, m, z_1. We want to decide whether z_1 is chosen at random or $z_1 = m^{x_1}$.

1. Choose x_2 at random and compute $h_2 = g^{x_2}$ and $z_2 = m^{x_2}$.
2. Execute the protocol $(P_2, \tilde{V})$.
3. If $\tilde{V}$ outputs P_0 then output 0. Otherwise output 1.

It is easy to see that $\tilde{V}$'s view in case $z_1 = m^{x_1}$ is that generated by P_0 and if z_1 is chosen at random it is the same as that generated by P_2. □

This lemma shows that given a signature from either P_1 or P_2 it is not feasible to tell which secret was actually used. However, in general the distinguisher may have received many signatures before trying to recognize which secret key was used in a given signature.

In order to say anything about this we assume that the hash function used in the signature scheme is such that making a signature equivalent to executing the basic proof system with an honest verifier (i.e., choosing the challenge using $\mathcal{H}$ corresponds to choosing the challenge at random).

Assume that a distinguisher can identify the signer of a signature given some previous signatures and the identity of the corresponding signers. Then by the above arguments it can be assumed that the distinguisher could also identify the signer after executing the protocol in Figure 6.4 acting as the honest verifier. However, given the identity of the prover, the transcript of the honest verifier can be generated with the correct distribution by the distinguisher itself. What the distinguisher can obtained from these previous signatures are $m_i^{x_j}$ for the various messages $m_1, m_2, \ldots$ and $j = 1$ or $j = 2$. Under the following assumption, this previous information has no help to the distinguisher.

Consider an oracle algorithm, A, on input (p, q, g, h_1, h_2) working as follows:

1. Repeat the following a polynomial number of times:
   (a) Choose a message m at random and select $j \in \{1, 2\}$.
   (b) Get m^{x_j} from the oracle.
2. Get a pair (m_0, z_0), where m_0 is chosen at random and z_0 is either $m_0^{x_1}$ or $m_0^{x_2}$.
3. Output j.

Assumption 3 For every polynomially bounded A as above, the probability that A outputs j such that $z_0 = m_0^{x_j}$ is "polynomially close" to $\frac{1}{2}$.

6.8 Conclusion

A formal definition for secure group signature proposed in this chapter includes properties: against forgery, (unconditional) anonymity, and signer identification. Based on witness hiding proofs of knowledge, a class of new group signature schemes are constructed. The schemes can provide unconditional or computational anonymity. The signers can be identified in an efficient and flexible way. Some theoretical results are proved. It is shown that in order to sign more messages with unconditional anonymity, the size of secret keys must grow. Also the auxiliary information must contain enough to identify the signer.

The scheme with computational anonymity is much more efficiency. But the proof of anonymity depends on some assumptions, which have not been proved equivalent to widely used Diffie-Hellman assumption and discret logarithm assumption (see [DH76]).

List of Figures

2.1 Basic three move protocol 20
2.2 Semi-proof of a square root of x modulo n 22

3.1 Diverting the semi-proof of a square root. 30
3.2 Three party protocol 31
3.3 Parallel transferability. 34
3.4 Transferring or diverting the basic protocol. 37
3.5 Notation for parallel divertibility. 38
3.6 Diverting a proof to many verifiers 48
3.7 Constructing of $\tilde{P}^*$ 49
3.8 Diverting the semi-proof of square root with different input . . 53
3.9 Convincing verifier and warden unconditionally. 56

4.1 Basic protocol: proof of $\log_g h = \log_U Z$ 61
4.2 Diverting proof of $\log_g h = \log_U Z$ 62
4.3 Transferring credentials 63

5.1 How a single mix works 72
5.2 Cascaded mixes 73
5.3 An encrypted ballot (on bulletin board) 74

6.1 Proving knowledge of n witnesses 94
6.2 Proving knowledge of one of n witnesses 95
6.3 Proving that z is constructed correctly with respect to h_1 . . . 97
6.4 Proving $z_1 = m^{\log_g h_1}$ or $z_2 = m^{\log_g h_2}$ 109

Bibliography

[BD91] M. Burrnester and Y. Desmedt. All Languages in NP Have Divertible Zero-Knowledge Proofs and Arguments under Cryptographic Assumptions. In *Advances in Cryptology - Proceedings of Eurocrypt' 90,* Lecture Notes in Computer Science #473, Springer-Verlag, 1991, pp. $1-10$

[Bran94a] S. Brands. Untraceable Off-line Cash in Wallet with Observers. In *Advances in Cryptology – Proceedings of CRYPTO 93.* Lecture Notes in Computer Science #773, Springer Verlag, 1994, pp. $302-318$.

[Bran94b] S. Brands. Untraceable Off-line Cash Based on the Representation Problem. Manuscript. To be published as a CWI Technical Report in January/February, 1994.

[BY86] J. Benaloh, M. Yung. Distributing the Power of a Government to Enhance the Privacy of Votes. In *Proceedings of the Fifih ACM Symposium on Principles of Distributed Computing,* pages $52-62$, 1986.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty Unconditionally Secure Protocols. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing,* pages $11-19$, 1988.

[CDG88] D. Chaum, I. Damgård, and J. van de Graaf. Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result. In *Advances in Cpyptology - proceedings of CRYPTO 87,* Lecture Notes in Computer Science #293, pages $87-119$. Springer-Verlag, 1988.

[CDS94] R. Cramer, I. Damgård and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology - proceedings of CRYPTO 94.*

[CE86] D. Chaum, J. H. Evertse. A Secure and Privacy Protecting Protocol for Transmitting Personal Information between Organizations. In *Advances in Cryptology - proceedings of CRYPTO 86,* Lecture Notes in Computer Science #263, pages 118−168. Springer Verlag, 1986.

[CF85] J. D. Cohen and M. J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science,* pages 372 − 382, 1985.

[CFN90] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Advances in Cryptology - proceedings of CRYPTO 88,* Lecture Notes in Computer Science #403, pages 319−327. SpringerVerlag, 1990.

[Ch81] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In *Commun. ACM,* **24**, 1981, pp. 84 − 88.

[Ch82] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology - Proceedings of Crypto '82,* Plenum Press, 1983, pp. 199 − 203.

[Ch91] D. Chaum. Zero-knowledge Undeniable Signatures. In *Advances in Cryptology - Proceedings of Eurocrypt '90,* Lecture Notes in Computer Science #473, Springer-Verlag, 1991, pp. 458 − 464.

[CheDa93] L. Chen and I. B. Damgård Security Bounds for Parallel Version of Identification Protocols. In *Advances in Cryptology - Proceedings of Eurocrypt '92,* Lecture Notes in Computer Science #658, Springer-Verlag, 1993, pp. 459 − 466.

[CheDaPe94] L. Chen, I. B. Damgård, T. P. Pedersen Parallel Divertibility of Proofs of Knowledge. In *Advances in Cyyptology - Proceedings of Eurocrypt '94.*

[ChePe94a] L. Chen and T. P. Pedersen New Group Signature Schemes. In *Advances in Cryptology - Proceedings of Eurocrypt '94.*

[ChePe94b] L. Chen and T. P. Pedersen Group Signatures: Unconditional Security for Members. *Technical Report,* Computer Science Department, Aarhus University.

[18] Glenn Bruns. A case study in safety-critical design. Technical Report ECS-LFCS-92-239, Laboratory for Foundations of Computer Science, University of Edinburgh, September 1992.

[CP92] D. Chaum and T. P. Pedersen. Wallet Databases with Observers. In *Advances in Cryptology - proceedings of CRYPTO 92,* Lecture Notes in Computer Science #740, pages $89-105$. Springer Verlag, 1993.

[CvH91] D. Chaum, E. van Heijst. Group Signatures. In *Advances in Cryptology - proceedings of EUROCRYPT 91,* Lecture Notes in Computer Science #547, pages $257-265$. SpringerVerlag, 1991.

[D93] Y. Desmedt. Threshold Cryptosystems. In *Advances in Cryptology - proceedings of AUSCRYPT 92,* Lecture Notes in Computer Science #718, pages $3-14$, 1993.

[Da88] I. B. Damgaard. Payment Systems and Credential Mechanisms with Provably Security Against Abuse by Individual. In *Advances in Cryptology - proceedings of CRYPTO 88,* Lecture Notes in Computer Science #403, pages $328-335$. Springer-Verlag, 1990.

[DGB88] Y. Desmedt, C. Goutier, and S. Bengio. Special Uses and Abuses of the Fiat-Shamir Passport Protocol. In *Advances in Cryptology - proceedings of CRYPTO 87,* Lecture Notes in Computer Science #293, pages $21-39$. Springer-Verlag, 1988.

[DH76] W. Diffie and M. E. Hellman New Directions in Cryptography. In *IEEE Trans. Inform.,* IT-22(6):$644-654$, November, 1976.

[FFS88] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge Proofs of Identity. *Journal of Cryptology,* 1(2):$77-94$, 1988.

[FOO92] A. Fujioka, T. Okamoto, K. Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *Advances in Cryptology - Proceedings of AUSCRYPT 92.* Lecture Notes in Computer Science #718, 1994.

[FS87] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Pre-proceedings of EUROCRYPT 86,* pages 186 – 194. 1987.

[FS90] U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing,* pages 416 – 426, 1990.

[GM84a] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences,* 28(2):270 – 299, 1984.

[GM84b] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences,* 28:270 – 299, 1984.

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attack. *SIAM Journal on Computing,* 17(2):281 – 308, April 1988.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-systems. *SIAM Journal of Computation,* 18(1):186 – 208, 1989.

[H92] E. van Heijst. *Special Signature Schemes.* PhD thesis, CWI, 1992.

[Iv92] K. R. Iversen. A Cryptographic Scheme for Computerized General Elections. In *Advances in Cryptology - Proceedings of Crypto '91,* Lecture Notes in Computer Science #576, SpringerVerlag, 1992, pp. 404 – 419.

[ISS93] T. Itoh, K. Sakurai, and H. Shizuya Any Language in IP Has a Divertible ZKIP. In *Advances in Cryptology - Proceedings of ASIACRYPT '91,* Lecture Notes in Computer Science #739, Springer Verlag, 1993, pp. 382 – 397.

[NY90] M. Naor and M. Yung. Public-key Cryptosystems Provoably Secure Against Chosen Ciphertext Attacks. In *Proceedings of the*

22-nd Annual ACM Symposium on Theory of Computing., pages: 427 – 437, 1990.

[Ok88] T. Okamoto. A Digital Multisignature Scheme Using Bijective Public-Key Cryptosystems. *ACM Trans. on Camp, Sys.,* 6(8): 432 – 441, 1988.

[OO89] T. Okamoto and K. Ohta. Divertible Zero Knowledge Interactive Proofs and Commutative Random Self-Reducibility. In *Advances in Cryptology - proceedings of EUROCRYPT 89,* Lecture Notes in Computer Science #434, pages 134 – 149. Springer-Verlag, 1990.

[OO90] K. Ohta and T. Okamoto. A Modification of the Fiat-Shamir Scheme. In *Advances in Cryptology - Proceedings of Crypto '88,* Lecture Notes in Computer Science #403, Springer-Verlag, 1990, pp. 232 – 243.

[OO93] K. Ohta and T. Okamoto. A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme. In *Advances in Cryptology - proceedings of ASIACRYPT 91,* Lecture Notes in Computer Science #739, pages 139 – 148. Springer-Verlag, 1993.

[PIK93] C. Park, K. Itoh, K. Kurosawa Efficient Anonymous Channel and All/Nothing Election Scheme. In *Advances in Cryptology - proceedings of EUROCRYPT 93,* Lecture Notes in Computer Science #765, pages 248 – 259. Springer-Verlag, 1993.

[P91] T. P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Advances in Cryptology - proceedings of EUROCRYPT 91,* Lecture Notes in Computer Science #547, pages 522 – 526. Springer-Verlag, 1991.

[P92] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology - proceedings of CRYPTO 91,* Lecture Notes in Computer Science #576, pages 129 – 140. Springer Verlag, 1992.

[Sch90] C. P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - proceedings of CRYPTO 89,* Lecture Notes in Computer Science, pages 239 – 252. Springer Verlag, 1990.

[Sim84] G. J. Simmons. The Prisoner's Problem and the Subliminal Problems. In *Advances in Cryptology - proceedings of CRYPTO 83,* Plenum Press, pages 51 – 67. 1984.

[Sm93] B. Schoenmakers. Efficient Proofs of Or. Manuscript, 1993.

[TW87] M. Tompa and H. Wall. Random Self-reducibility and Zero Knowledge Interactive Proofs of Possession of Information. In *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science,* pages 472 – 482, 1987.