VERIFIABLE DEMOCRACY

Yvo Desmedt
desmedt@cs.uwm.edu
Dept. of Elec. Eng. & Comp. Science
University of Wisconsin-Milwaukee
PO Box 784, Milwaukee, WI 53201, USA
Tel. +1-414-229-6762, Fax+1-414-229-6958, and
Center of Cryptography Computer and Network Security
University of Wisconsin-Milwaukee, and
Department of Mathematics
Royal Holloway
University of London, U.K.

Brian King
bking@allmalt.cs.uwm.edu
Dept. of Elec. Eng. & Comp. Science
University of Wisconsin-Milwaukee

Abstract In a k out of n threshold signature scheme the secret key is distributed to n participants, so that any subset B of participants, with $|B| \geq k$, can combine their shares to form a signature, while any subset of cardinality $\leq k-1$ gain no information about the signature. In democratic organizations the number of users vary temporally while maintaining the relationship $k = \lfloor \frac{n}{2} \rfloor + 1$. The manner in which a legislature votes is similar to a threshold signature scheme, and the power to sign is similar to possessing shares to sign. The transfer of power to sign is an integral part of democracy. In recent work, redistribution schemes have been developed that allow one to vary the threshold k and the number of users n. However, these solutions require parties to delete their shares, which is often an unrealistic assumption. Here we provide a model for democratic bodies and solve the related problem of assuring an orderly and verifiable transfer of power as the size of the body varies.

Keywords: Applied Cryptography, Authentication, Cryptographic Protocols, Threshold Signatures, Verifiable Signature Sharing

B. Preneel (ed.), *Secure Information Networks*

1. INTRODUCTION

A threshold cryptosystem can be used for those situations where at least one of the parties is an organization rather than an individual. To achieve this, a secret needs to be distributed throughout the organization so that authorized groups in the organization can *use* the secret, and ensure that no unauthorized group gains any information about the secret. For example, In the banking industry, large transactions need to authenticated. If the authentication of the transaction was left to an individual's discretion, then this could lead to invitation to embezzlement. The need to consider such scenarios was observed by Shamir in [12]. Thus authentication could be shared out to group(s) within the bank.

Threshold cryptosystems are based on threshold schemes. In a k out of n threshold scheme [12, 1] all subsets of cardinality $\geq k$ are authorized sets and can recover the secret, and all subsets of cardinality $\leq k-1$ are unauthorized sets who cannot recover the secret.

A topic related to our interest is "dynamic threshold sets". Can threshold schemes be developed which will allow a threshold or the number of participants to vary during its lifetime? The problem "how participants authorized by their access structure Γ, can redistribute shares to a new set of participants in a new access structure", was addressed in [6, 8]. In [6, 8], redistribution protocols were developed that will allow a set of n participants in a k out of n threshold scheme to redistribute their shares to n' participants so that any k' out of n' can recover the secret key.

This paper will address a problem related to "dynamic threshold sets". Consider the following: A legislative body, like a congress, often passes "laws" according to some minimum of yes votes which is a proportion of the body present (some possibilities include majority or two-thirds). In such a situation the threshold is dependent on a proportion of the body that is present on a given day[1]. One day it may be a 51 out of 100 threshold, the next day a 45 out of 88. Consequently, we would need a scheme which allows transfer[2] of signature power.

The goal here is to provide an accurate model for democratic bodies and solve the related problem of assuring an orderly and verifiable

[1]The unit of time is relative. Most legislatures are dynamic throughout a given day, so we could substitute any time interval for "day ".

[2]We use the word transfer as an analogy, the participant will redistribute their share to the n' participants.

transfer of power as the size of the body varies. Let us first differentiate between situations when the democratic body has been terminated and when the body is being dynamic in size. For example, for situations when some type of election has taken place, or if a member has been thrown out of office for some reason, we would view that the body has been terminated, and that a new one has begun. The body is being dynamic when membership doesn't change but the threshold has changed. For example, the body's threshold may go from a two-thirds majority to a simple majority (like in the U.S. Senate), or the body maybe working with a threshold of a simple majority, and the threshold is changing due to absenteeism. It is the latter "dynamic threshold" that we will address. Thus when many members are removed we will refer to this as the body has been terminated, and its term has expired. Any time the body is able to consider motions, we will refer to this as the body is in session. Any time when the threshold of the body has changed, yet the body has not been terminated, we will refer to this as a session has ended and a new one will begin. As we will discuss later, when one considers different legislative bodies, there is a great diversity. Some bodies are not dynamic at all, whereas others are in constant change. Some bodies have only one session in their term, and others can have multiple sessions in a given day. We would like to develop a model for a democratic body [3] for secure transfer of power of signing when session changes occur. Further, we will view that members of the body may be adversarial, and that the body is an electronic body, because of this verification of the transfer is important. This model will be used as a guide to develop verifiable democracy.

Consider a legislative body $P_1, \ldots, P_n$. They possess shares to a secret key which allows motions presented to the body, to be voted on, passed, and signed into law. The number of legislators present will vary from time to time. As long as a quorum exists (a pre agreed minimum number of legislators needed to be present), the motions can be passed, according to some threshold, for simplicity we will adopt a simple majority vote. i.e. a k_t out of n_t where n_t represents the number of legislators present at time t, and $k_t = \lfloor \frac{n_t}{2} \rfloor + 1$. Again for the sake of simplicity, let's assume that the quorum represents a majority of the total number of legislators, i.e. the quorum is $k = \lfloor \frac{n}{2} \rfloor + 1$. At any time there may be n_t legislators

[3]The use of legislators is merely an analogy, one could substitute any body whose threshold set is dynamic, a board of directors, the set of citizens attending a town hall meeting, or even the set of faculty attending a faculty meeting.

present, a motion can be voted on as long as $n_t \geq k$. In that case the motion would be voted on in a k_t out of n_t threshold scheme, where $k_t = \lfloor \frac{n_t}{2} \rfloor + 1$. Before the legislators $P'_1, \ldots, P'_{n_t}$ vote on the motion, a set $P_{i_1}, \ldots, P_{i_k}$, which is a subset of $P'_1, \ldots, P'_{n_t}$, will have to send to the legislators which are present, i.e. $P'_1, \ldots, P'_{n_t}$, the power to use the secret key to sign. They can do this since they possess shares of the key in a k out of n threshold scheme.

First observe that the transfer of signature power must be temporary. For example, suppose that the original body is of size 100, then one day only 88 are present. Any 51 out of the 88 participants can transfer their signature power to obtain a 45 out of 88 signature scheme, so that 45 can sign. If the next day there are 97 present, then the threshold has now changed to 49, and 45 should not be able to sign.

Consider some attempts at a solution of redistributing the power of signing. Every time a new threshold is needed we simple create a new secret key. The cost, however, is prohibitive. This cost can be measured in time needed to authenticate new public keys, as well as, the cost of publishing new public keys. Another reason why this is not an appropriate solution is that it is not natural. The secret key should be associated with the legislative body, and only when the term of this body is completed, should a new key be established. In addition, absenteeism may be a common feature of the legislature. For example, consider the town hall meeting example, in this situation it is one-person one vote, during a single meeting citizens may walk in and/or walk out. Each time a new key would need to be generated.

A second attempt at a solution is to use the results developed by [6, 8]. This would allows us to go from a k out of n threshold scheme to a k' out of n' threshold scheme. In these schemes, participants must destroy their old shares. In some applications this can be an acceptable assumption, but as a requirement it invites abuse as we now discuss. For example, suppose that the body is of size 100, shrinks to 88, and then grows to 97. Since the set of participants has shrunk to 88, the threshold has shrunk to 45. When the set of participants grows to 97, the threshold has grown to 49. To prevent 45 members (of the old threshold) to be able to sign, 97-45+1=53 members must delete their old shares. Which is larger than the original majority! And if they do not, then 45 members can sign! This is contrary to our intentions.

Let us describe some properties that a complete solution should possess. We have already observed that the transfer of signature power should be temporary. This can be achieved by having participants

$P_{i_1}, \ldots, P_{i_k}$ *transfer their partial signatures instead of their power to sign.*

Secondly, observe that a few of the k (out of the n_t) participants $P_{i_1}, \ldots, P_{i_k}$ could thwart the process by not properly transferring their power (shares). This would be especially true if the message (law) was such that they had a vested interest that the law should not be passed. Thus, as the transfer of power is message oriented, there is a need for the set $P_{i_1}, \ldots, P_{i_k}$ to transfer power blindly (i.e. without seeing the message).

Third, the participants $P'_1, \ldots, P'_{n_t}$, when given an opportunity to act on legislation must know that the outcome ("sign" or "not sign") is a result of *their* decision and not a result of bad faith on the part of the participants $P_{i_1}, \ldots, P_{i_k}$ who had transferred them the power to sign. For example, if the law was not signed, who is to say that this is because there did not exist k_t participants, a majority in $P'_1, \ldots, P'_{n_t}$, who wanted to sign or rather that a majority in $P'_1, \ldots, P'_{n_t}$ were not ever truly given the power to sign that message temporarily blinded.

Lastly, no set of participants should gain any information about a motion made during an illegal session, a session where either cheaters have been discovered or the number of legislators present is less than the quorum k.

In light of the fact that we intend to verify transfer, one may wonder whether transferring this power blindly is still a necessary requirement. However, for various reasons we think that this aspect of the model is essential. Let us consider some examples of legislative bodies. A President of the U.S. is an example of a 1 out of 1 threshold. There is no dynamic aspect to this threshold. The session runs the length of the term, which starts when the President takes office and ends when the next president takes office. Legislation is blinded to the President when the transfer of power occurs. That is, there is no revealing of all possible legislation to the President-elect at the moment of transfer occurs. In a town hall meeting or board of directors, any member at that moment may make a motion. The session had started once the group of people had congregated for the meeting. Thus the motion has been made public after the transfer of power has occurred. Most government legislatures have formed committee-based legislation. A proposed law, may be voted on to send it to a committee, it works its way through committee to see if it is prudent, then making its way back to the legislature for a final vote. This structure exists because there is a realization that some laws need to be carefully considered until a final vote. However, the original

motion was blinded, and a vote occurs to determine whether it should be sent to a committee. Arguably the initial motion was blind in the sense that the transfer of power has already occurred when the legislators congregate together, so the transfer occurs without seeing what motions will be voted on.

An important tool in politics is delay. As we have detailed motions are made after the transfer of power has occurred. A model which requires motions to be revealed before the transfer of power has taken place lends itself to encouraging deceit, because delay is a useful tool. True our scheme will employ verification of the transfer, but history has shown how one member is willing to sacrifice their career for the good of the political coalition that they belong to. That is, if a coalition realizes that the number of coalition members is not sufficient to block passage (or to ensure passage) of a motion a member may cheat for the good of their coalition by sending false shares, it is detected because of the verification (and so the member may be penalized because of it). But it is effective, because the vote on the motion has been delayed.

Because we are considering a electronic legislative body, there is no "physical presence" which will represent transfer of power, hence in our model we will adopt a need of motions/laws be blinded prior to given to the body, after the transfer occurs the motion will be revealed.

We start with a k out of n threshold scheme. At time t, m_t will represent the message. The following will describe a protocol which allows temporary transfer of signature power from a set of k participants $P_{i_1}, \ldots, P_{i_k}$ to a set of n_t participants $P'_1, \ldots, P'_{n_t}$. Further, the protocol includes verification, so that the n_t participants can verify that the k participants have honestly sent them this power. Most important, the n participants never cede power of future signatures to the n_t participants $P'_1, \ldots, P'_{n_t}$. The protocol can be applied to any RSA threshold scheme [11], which allows transfer of partial signatures, as long as the threshold scheme described by the partial signatures is a linear secret sharing scheme.

2. DEFINITIONS AND MODEL

Definition 2..1 *A k out of n threshold scheme [1, 12] consists of two algorithms: a distribution algorithm $\mathcal{D}$ and a reconstruction algorithm $\mathcal{R}$ such that if $P_1, \ldots, P_n$ are n participants, then for each $\mathbf{s} \in \mathcal{K}$ (the set of secrets), shares $s_1, \ldots, s_n$ are constructed from the sets $\mathcal{S}_1, \ldots, \mathcal{S}_n$ (called the share sets), using distribution algorithm $\mathcal{D}$, and distributed to the participants $P_1, \ldots, P_n$, respectively, such that any k out of n*

participants can, using their shares and the reconstruction algorithm $\mathcal{R}$, compute **s**. *Further, any set B' of participants, with $|B'| \leq k-1$, gain no information about the* **s**. *We will denote this k out of n threshold scheme by $(\mathcal{D}, \mathcal{R})$. The set of all B, for which $|B| \geq k$, is denoted by Γ.*

2.1 LINEAR SECRET SHARING SCHEME

Definition 2..2 *[6] The class of secret sharing schemes in which the key* $\mathbf{s} = \sum_{i \in B} \Psi_{i,B}(s_i)$ *where $B \in \Gamma$ and $\Psi_{i,B} : S_i(+) \rightarrow \mathcal{K}(+)$ is a homomorphism, for each $i \in B$, is called a linear secret sharing scheme.*[4]

Karnin, Greene and Hellman [9] were the first to consider such schemes, their work was concentrated in the area of Galois fields. Some examples of recent work in the area of linear secret sharing occurs in [7, 6, 8].

2.2 RSA SIGNATURE SCHEME

The RSA signature scheme is an example of a linear secret sharing scheme, and will be used to create a solution to the verifying democracy problem.

Suppose Alice wishes to create a public key using RSA. Alice randomly selects two distinct primes, of sufficient size. Alice forms N which is the product of the two primes. Let $\phi(N)$ denote the Euler totient function. To form a RSA public key, Alice randomly selects $e \in_R Z^*_{\phi(N)}$, whereupon Alice computes d such $ed \bmod \phi(N) = 1$. Alice publishes N and e, and keeps d as her private key.

Suppose Alice wishes to send to Bob a signature of message M. Alice applies a hash function $h()$ to M, so that $m = h(M)$. Alice sends to Bob M and m^d, whereupon Bob can verify the signature by computing $m = h(M)$ and raising it to the power e. If the result of the signature is m then Bob accepts the message.

2.3 THRESHOLD RSA SIGNATURE SCHEME

The problem of extending the RSA signature scheme to a threshold signature scheme is that $\phi(N)$ cannot be revealed to any of the par-

[4]Thus a linear secret sharing scheme is a threshold scheme such that for all $B \in \Gamma$, $\mathcal{R}(\vec{s_B}) = \sum_{i \in B} \Psi_{i,B}(s_i)$.

ties, even to the n participants $P_1, \ldots, P_n$. Threshold RSA schemes are described in [5, 10].

Set up The secret is $d \in Z^*_{\phi(N)}$. A "trusted dealer" computes shares $s_1, \ldots, s_n$ and sends them privately to $P_1, \ldots, P_n$, respectively, so that for all $B \subset \{P_1, \ldots, P_n\}$, with $|B| = t$, $\sum_{i \in B} \Psi_{i,B}(s_i) = d$. In [5], the $\Psi_{i,B}$ represents a row matrix of nonnegative integers. Therefore there exists coefficients $a_{i,B}$ such that $d = \sum_{i \in B} a_{i,B} s_i$.

Now if M is some message, then $M \in Z^*_N$. Apply an appropriate hash function $h()$ to M resulting in $m = h(M)$.

Signature The RSA threshold scheme[5] is a linear secret sharing scheme. Thus as $\sum_{i \in B} \Psi_{i,B}(s_i) = d$, we have

$$m^d = m^{\sum_{i \in B} a_{i,B} s_i} = \prod_{i \in B} m^{a_{i,B} s_i} = \prod_{i \in B} S_i^{a_{i,B}}.$$

2.4 REDISTRIBUTION

To explain how redistribution works let us assume that Ψ represents a matrix. Suppose

$$\Psi_B = [\Psi_{i_1 B}, \Psi_{i_2 B}, \ldots, \Psi_{i_k B}]$$

where $B = \{P_{i_1}, \ldots, P_{i_k}\}$,

$$\mathbf{s} = [\Psi_{i_1,B}, \Psi_{i_2,B}, \ldots, \Psi_{i_k,B}] \begin{bmatrix} s_{i_1} \\ s_{i_2} \\ \vdots \\ s_{i_k} \end{bmatrix}$$

and where $\mathbf{s}$ is the secret key.

For each i_j, $j = 1$ to k, P_{i_j} will distribute s_{i_j} to $P'_1, \ldots, P'_{n_t}$ such that for all B', with $|B'| = k_t$:

$$s_{i_j} = [\Psi'_{1,B'}, \Psi'_{2,B'}, \ldots, \Psi'_{n_t,B'}] \begin{bmatrix} s'_{i_j 1} \\ s'_{i_j 2} \\ \vdots \\ s'_{i_j n_t} \end{bmatrix}$$

[5]The notation for threshold RSA has been simplified, for the precise definition of exponentiation see [5].

(observe that for each B' there are precisely k_t of the $\Psi_{l,B'}$ which are non zero, the remaining are zero). Then

$$[\Psi'_{1B'}, \Psi'_{2B'}, \ldots, \Psi'_{n_tB'}] \begin{bmatrix} s'_{i_1 1} & s'_{i_2 1} & \cdots & s'_{i_k 1} \\ s'_{i_1 2} & s'_{i_2 2} & \cdots & s'_{i_k 2} \\ \vdots & \vdots & \cdots & \vdots \\ s'_{i_1 n_t} & s'_{i_2 n_t} & \cdots & s'_{i_k n_t} \end{bmatrix} = [s_{i_1}, s_{i_2}, \cdots, s_{i_k}].$$

Therefore for any B', and B fixed,

$$\mathbf{s} = [\Psi'_{1,B'}, \Psi'_{2,B'}, \ldots, \Psi'_{n_t,B'}] \begin{bmatrix} s'_{i_1 1} & s'_{i_2 1} & \cdots & s'_{i_k 1} \\ s'_{i_1 2} & s'_{i_2 2} & \cdots & s'_{i_k 2} \\ \vdots & \vdots & \cdots & \vdots \\ s'_{i_1 n_t} & s'_{i_2 n_t} & \cdots & s'_{i_k n_t} \end{bmatrix} \begin{bmatrix} \Psi_{i_1,B} \\ \Psi_{i_2,B} \\ \vdots \\ \Psi_{i_k,B} \end{bmatrix}.$$

Hence

$$\tilde{S}_l = \begin{bmatrix} s'_{i_1 l} & s'_{i_2 l} & \cdots & s'_{i_k l} \end{bmatrix} \begin{bmatrix} \Psi_{i_1,B} \\ \Psi_{i_2,B} \\ \vdots \\ \Psi_{i_k,B} \end{bmatrix}$$

(observe that $\tilde{S}_l$ is independent of B'). So for any B'

$$\mathbf{s} = [\Psi'_{l_1}, \Psi'_{l_2}, \ldots, \Psi'_{l_{k_t}}] \begin{bmatrix} \tilde{S}_{l_1} \\ \tilde{S}_{l_2} \\ \vdots \\ \tilde{S}_{l_{k_t}} \end{bmatrix}$$

where $B' = \{P'_{l_1}, \ldots P'_{l_{k_t}}\}$. Thus $\tilde{S}_l$ represents the share (called the "compressed share") of P'_l which has been created by the redistribution of the shares $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$

The transfer of signature power will be performed by redistributing *partial signatures*, and *not* shares of d, in a k out of n threshold scheme to partial signatures in a k_t out of n_t threshold scheme, using the redistribution protocol in [6]. We will omit further details of the description of the redistribution protocol, and refer the reader to [6, 8].

2.5 VERIFIABLE SIGNATURE SHARING SCHEME

Verifiable signature sharing is a cryptographic sharing technique which allows a holder of document to distribute shares of the signature of the

document to proxies (participants), so that the proxies can later reconstruct and sign the document (if they wish). Further, by the end of the distribution phase, honest proxies can verify that they have been given shares of the authentic signature, without reconstructing the signature.

2.6 MODEL

We start with a k out of n threshold scheme. At time t, n_t will represent the number of participants present, m_t will represent the message, and k_t will represent the dynamic threshold. A quorum exists provided $n_t \geq k$. Whenever $n_t \geq k$, we assume the following: (i) $k_t < k$ and (ii) $k_t \leq n_t$. In the case $n_t < k$, we set k_t to ∞. Let us denote $A_t = \{P'_1, \ldots, P'_{n_t}\}$ as the set of participants present at time t. B_t, a subset of A_t, represents the set of participants present at time t which are willing to sign m_t.

MODEL 2.6

(i) (completeness) *If $n_t \geq k$ then $\forall B_t$, with $|B_t| \geq k_t$, either B_t can sign m_t or they can identify cheaters.*
(ii) (soundness) *If $B'_t \not\subset A_t$ or if $|B'_t| < k_t$ then B'_t cannot sign $m_t \notin \{m_i : 1 \leq i \leq t-1\}$*
(iii) The action of the cheaters should be independent of the message. Therefore if $\forall B'$ (represents a set of cheaters), with $|B'| < k$, $\forall m_t, m'_t$ the family $\{Joint-View_{B'}(x, h, m_t)\}_x$ is computationally indistinguishable from $\{Joint-View_{B'}(x, h, m'_t)\}_x$, where x is the RSA public key and h is the history.
(iv) If $n_t < k$ or if cheaters have been discovered, then no subset of A_t should gain any information about m_t. Therefore $\forall m_t, m'_t$ the family $\{Joint-View_{A_t}(x, h, m_t)\}_x$ is computationally indistinguishable from $\{Joint-View_{A_t}(x, h, m'_t)\}_x$

3. THE MAIN TOOL

We will be working in an RSA setting. Observe that the secret key d, represents the true signature power. If the participants $P_{i_1}, \ldots, P_{i_k}$ redistribute their shares of d to $A_t = \{P'_1, \ldots, P'_{n_t}\}$ to sign some message m_t, then this set A_t possesses the power to sign future messages. To expect the set A_t to erase their shares of d, is not realistic. Thus what should be transferred is *shares of the signed message* not *shares of d* (shares of the signed message are called "partial signatures").

As we have mentioned, the message will be blinded. We can achieve a "blinded m_t" by raising m_t to the power e to form $m_t^e \bmod N$ (here the e refers to the exponent which is public in the RSA scheme). Now if shares of d were provided to $P_1, \ldots, P_n$, then the partial signatures applied to m_t^e would actually be partial signatures of $m_t^{ed} = m_t$, which is not a signature of m_t (we would like m_t^d). We solve this problem by distributing shares of d^2 to $P_1, \ldots, P_n$ rather than shares of d. Then when we were to apply partial signatures to m_t^e, they are shares of $m_t^{ed^2} = m_t^d$.

4. PROTOCOL – A DEMOCRATIC THRESHOLD SCHEME

The following protocol describes a democratic threshold scheme that solves the posed problem.

D will represent a "dealer". D will be involved in generating N, public key e and secret key d. Then D publishes (e, N). D will also be responsible for creating the original shares of d^2, and distributing them privately to $P_1, \ldots, P_n$.

Since (e, N) is public, any person (even people who are not legislators) can make a motion M_t to the legislative body. Before they present the motion, they will apply a hash function $h()$ to it. The result is $m_t = h(M_t)$. Then they blind m_t and M_t, and present m_t^e to the legislators which are currently present, i.e. $A_t = \{P'_1, \ldots, P'_{n_t}\}$. This person, doing the blinding, will be denoted by $\mathcal{B}$. Throughout the protocol we assume that if a cheater is discovered, or if a COMPLAINT is registered, then the participants will terminate the protocol (or they will complete the step that identifies the cheater, and then terminate). We assume that $k_t \geq \lfloor \frac{n}{2} \rfloor + 1$ (i.e. k_t is greater than or equal to a majority).

From now on, shares that are represented by lower case letters will denote shares of the secret key. Shares that are represented by upper case letters will denote partial signatures. Compressed shares will be denoted with a tilde. Coefficients used in a k out n will be given by $a_{i,B}$, whereas, coefficients used in a k_t out n_t will be given by $a'_{i,B}$.)

Initialization - I

I - 1. D chooses p and q primes such that $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' are primes, and sets $N = pq$. D chooses $e \in_R Z^*_{\phi(N)}$. d is determined such that $ed = 1 \bmod \phi(N)$. D makes (e, N) public D then creates shares s_i of d^2 using a RSA threshold scheme which is a linear secret sharing scheme. D privately

distributes the shares $s_1, s_2, \ldots, s_n$ to participants $P_1, \ldots, P_n$, respectively. Consequently, for all $B \in \Gamma$

$$\sum_{i \in B} a_{i,B} s_i = d^2 \bmod \phi(N)$$

I - 2. D chooses a "test" message $w \in Z_N^*$, and publishes $(w, w^{s_i e^2}) = (w, W_i)$ for each $i = 1, \ldots, n$. Observe that $w^{s_i e^2}$ is a share of w, which can be computed in a k out of n manner.

Transfer of motion -TM

TM - 1. An individual $\mathcal{B}$ makes a motion M_t, and applies the hash function $h()$ to it. The result is $m_t = h(M_t)$.

TM - 2. $\mathcal{B}$ blinds m_t by computing $m_t \rightarrow m_t^e$ and similarly $E(M_t)$ is the blinding of M_t.

TM - 3. $b_t = m_t^e$ is then broadcasted to $A_t = \{P_1', \ldots P_{n_t}'\}$.

Transfer of signature shares - TS

TS - 1. If $n_t \geq k$ continue, otherwise they terminate the protocol.

TS - 2. Any k participants are chosen from the set A_t. We will denote them by $P_{i_1}, \ldots, P_{i_k}$.

TS - 3. For each j, participant P_{i_j} possess a share of d^2 which is denoted by s_{i_j}. Hence they possess shares of $(m_t^e)^{d^2}$. Such a share will be denoted by $S_{i_j} = (m_t^e)^{s_{i_j}}$.

TS - 4. P_{i_j} forms n_t shares of s_{i_j} using redistribution protocol described in [6]. These n_t shares will be denoted by $\tilde{s}_{1,i_j}, \ldots, \tilde{s}_{n_t,i_j}$. The result is a k_t out of n_t threshold scheme which will compute s_{i_j}.

TS - 5. For each $l = 1, \ldots, n_t$, P_{i_j} forms $\tilde{S}_{l,i_j} \stackrel{\text{def}}{=} (m_t^e)^{\tilde{s}_{l,i_j}}$ which is a share of S_{i_j}.

Recall $(m_t^e)^{\tilde{s}_{1,i_j}}, (m_t^e)^{\tilde{s}_{2,i_j}}, \ldots, (m_t^e)^{\tilde{s}_{n_t,i_j}}$ are k_t out n_t threshold shares which can compute S_{i_j}.

$$S_{i_j} = \prod_{\mu \in \tilde{B}} (m_t^e)^{\tilde{s}_{\mu,i_j} a'_{\mu,\tilde{B}}} = \prod_{\mu \in \tilde{B}} \tilde{S}_{\mu,i_j}^{a'_{\mu,\tilde{B}}}.$$

TS - 6. P_{i_j} sends $\tilde{S}_{l,i_j}$ to P_l' via private channels for each l, $l = 1, \ldots, n_t$

TS - 7. For each l, $l = 1, \ldots, n_t$, P_l' has received shares $\tilde{S}_{l,i_1}, \ldots, \tilde{S}_{l,i_k}$ which they compress according to the redistribution protocol in [6] to form a single share $\tilde{S}_l$.

Verification stage - VS one

VS one - 1. P_{i_j} broadcasts $S_{i_j}^{e^2}$ and $\tilde{S}_{1,i_j}^{e^2}, \tilde{S}_{2,i_j}^{e^2}, \ldots, \tilde{S}_{n_t,i_j}^{e^2}$ to $P'_1, \ldots, P'_{n_t}$ for $j = 1, \ldots k$

VS one - 2. The set A_t collectively verify that for each $l = k_t, k_t + 1, , \ldots, n_t$ that

$$S_{i_j}^{e^2} \stackrel{?}{=} \prod_{\mu \in C_l} (\tilde{S}_{\mu,i_j}^{e^2})^{a'_{\mu,C_l}}$$

where $C_l = \{1, 2, \ldots, k_t - 1, l\}$. If for any such l, this equality does not hold then P_{i_j} is a cheater.

VS one - 3. We now use the protocol described in [10] which is a generalization of [3]. P_{i_j} is the PROVER and the set A_t act as the VERIFIER. PROVER proves that $W_{i_j} = w^{s_{i_j} e^2}$ has been raised to the same power as the broadcasted message $(m_t^e)^{s_{i_j} e^2}$ where w, W_{i_j}, m_t^e are public (and have been determined in the initial stages).

1. VERIFIER chooses $\rho, \tau \in_R \{0, 1, \ldots, n\}$ and computes $R \stackrel{\text{def}}{=} (m_t^e)^\rho w^\tau$. VERIFIER commits to ρ, τ. VERIFIER sends this commitment and R to the PROVER.
2. PROVER computes $S_R \stackrel{\text{def}}{=} R^{e^2 s_{i_j}}$ and PROVER sends a commitment of S_R to the VERIFIER.
3. PROVER opens the commitment to ρ, τ.
4. VERFIER opens the commitment to S_R.
5. VERIFIER verifies that $S_R \stackrel{?}{=} ((m_t^e)^{s_{i_j} e^2})^\rho W_{i_j}^\tau$ (where the former was the broadcasted message).
6. If true VERIFIER accepts, otherwise VERIFIER broadcasts a (COMPLAINT, i_j). That is, i_j is a cheater.

VS one - 4. Each participant P'_l verifies that the privately sent share $\tilde{S}_{l,i_j}$ (from step **TS - 6**) raised to the e^2 power is the same as the broadcasted $\tilde{S}_{l,i_j}^{e^2}$ (which was broadcasted in **VS - one 1**). If not, then P'_l broadcasts a (COMPLAINT, l, i_j). If such a complaint is broadcasted then either P_{i_j} is a cheater or P'_l is a cheater. To determine which participant P_{i_j} or P'_l is a cheater perform the following protocol.

1. We assume that P'_l was privately sent $\tilde{S}_{l,i_j}$ (in step **TS - 6**) via a public key encryption, such that the ciphertext was public. Hence to prove that P_{i_j} was the cheater, P'_l signs $\tilde{S}_{l,i_j}$ and broadcasts the signature to A_t. (Failure of P'_l to perform this step will be considered as proof that P'_l is a cheater.)

2. The set A_t collectively decide (in a k_t out of n_t way) whether the signature reveals a message when raised to the e^2 power modulo N is the same as the message broadcasted by P_{i_j}. $\tilde{S}^{e^2}_{l,i_j}$. If it is then P'_l cheated, otherwise P_{i_j} cheated.

Verification stage - VS two

VS two - 1. We now use another protocol described in [10]. For each j, participant P_{i_j} broadcasts new test messages using the shares created in step **TS - 4**. They will be denoted by $(w^{e^2})^{\tilde{s}_{1,i_j}}, (w^{e^2})^{\tilde{s}_{2,i_j}}, \ldots, (w^{e^2})^{\tilde{s}_{n_t,i_j}}$. In a k_t out of n_t these messages should compute the test message $W_{i_j} = (w^{e^2})^{s_{i_j}}$.

VS two - 2. The set A_t collectively verify that for each $l = k_t, k_t + 1, \ldots, n_t$ that

$$W_{i_j} \stackrel{?}{=} \prod_{\mu \in C_l} ((w^{e^2})^{\tilde{s}_{\mu,i_j}})^{a'_{\mu,C_l}}$$

where $C_l = \{1, 2, \ldots, k_t - 1, l\}$. If for any l, this equality does not hold then P_{i_j} is a cheater.

VS two - 3. P_{i_j} is the PROVER and the set A_t collectively act as the VERIFIER. the PROVER proves that for each l, the broadcasted $(w^{e^2})^{\tilde{s}_{l,i_j}}$ is indeed an element of the group $\langle w^{e^2} \rangle$ as in [4]. The PROVER selects a random exponent r (Since the PROVER does not know $\phi(N)$, they use N, also due to the nature of threshold RSA (see [5]) the power is really a vector of integers, not an integer) and broadcasts $w' = (w^{e^2})^r$. The VERIFIER chooses a random bit b, If $b = 1$ then PROVER broadcasts the value $r + \tilde{s}_{l,i_j}$, and the VERIFIER verifies that $(w^{e^2})^{r+\tilde{s}_{l,i_j}} = w'(w^{e^2})^{\tilde{s}_{l,i_j}}$. If $b = 0$ then the PROVER broadcasts r, and the VERIFIER verifies that $(w^{e^2})^r = w'$. If the broadcasted $(w^{e^2})^{\tilde{s}_{l,i_j}} \notin \langle w^{e^2} \rangle$, then the probability of passing this test is $1/2$. By repeating this procedure ν times, we reduce this probability to $2^{-\nu}$.

VS two - 4. Each participant P'_l compresses the verified $(w^{e^2})^{\tilde{s}_{l,i_1}}, (w^{e^2})^{\tilde{s}_{l,2}}, \ldots, (w^{e^2})^{\tilde{s}_{l,i_k}}$ to $\tilde{W}_l$. Observe that with high probability both $\tilde{W}_l$ and $\tilde{S}_l$ belong to groups $\langle w^{e^2} \rangle$ and $\langle m_t^e \rangle$ such that the exponents are the same.

Act on legislation - AL

AL - 1. The person $\mathcal{B}$ who made the motion reveals m_t to A_t

AL - 2. The set A_t of participants verify that b_t is m_t^e

AL - 3. The k_t out of n_t legislative body A_t decide whether to sign or not to sign the legislation. Any subset of size k_t can sign the legislation, Since for all $B_t \subset A_t$, with $|B_t| = k_t$

$$\prod_{l \in B_t} \tilde{S}_l^{a'_{l,B_t}} = m_t^d \bmod N$$

AL - 4. To verify partial signatures for each $l \in B_t$ we apply the protocol in [10] which we used in **step VS - 3** using $\tilde{W}_l$ as the test message and $\tilde{S}_l$ as the partial signature that needs to be verified. Participant P'_l plays the role of the PROVER and the set A_t collectively act as the VERIFIER. This protocol will identify any cheaters.

AL - 5. If there are no cheaters, and $|B_t| \geq k_t$, the set B_t signed m_t.

5. REMARKS CONCERNING THE VERIFIABLE DEMOCRACY PROTOCOL

Let us consider our protocol in terms of the described model **Model 2.6**.

(i) Consider *completeness.* Observe that there are limitations to the ability to identify cheaters. First, if the majority of the n_t participants are cheaters, then cheaters may not be identified. However, if $k_t \geq \lfloor \frac{n_t}{2} \rfloor + 1$, and there are at least k_t honest participants then the protocol satisfies *completeness.*

(ii) Consider *soundness.* It is true that if B'_t is any set of people and if $|B'_t| < k_t$ then B'_t cannot sign m_t. However, if B'_t represents any set of people (not a subset of A_t), it is possible that B'_t can sign m_t. This is true if B'_t contains at least k participants from the original legislature. This follows from the fact that they possess shares to d^2. However, this violation of soundness is the only possible violation, and this seems to be an acceptable violation since a signature occurs only because of overwhelming support (i.e. k legislators willing to sign.) One could argue that if participants from the original legislature learn about a motion and vote on it then they are present (i.e. they belong to A_t.)

(iii) & (iv). We will not be able to achieve security condition (*iii*) of **MODEL 2.6**. That is the view of cheaters is such that even if they are discovered they will see m_t^e, which is an RSA encryption of m_t. However, the only information leaked is equivalent to the information leaked when one uses RSA to encrypt.

6. UNCONDITIONAL PRIVACY

We have achieved a verifiable version of democracy. However, we will not be able to achieve condition (*iii*) of **MODEL 2.6** unconditionally. The view of cheaters is such that they will see m_t^e, which is an RSA encryption of m_t, and of course information about m_t leaks. So we alter the scheme one last time, as follows (we will only indicate the steps that will change).
We alter steps **TM - 2** and **3** as follows:

> $\mathcal{B}$ blinds m_t by m_t^e. $\mathcal{B}$ also chooses $r \in_R \mathbf{Z}_N^*$. $\mathcal{B}$ sends $b_t = m_t \cdot r^{e^2}$ to A_t.

Observe that the mapping $x \longrightarrow x^{e^2}$ is an isomorphism from $\mathbf{Z}_N^*$ to $\mathbf{Z}_N^*$. Therefore if r is uniform random, then so is r^{e^2}. Note that the partial signatures formed by the set $\{P_{i_1}, \ldots, P_{i_k}\}$ will be for $m_t^e \cdot r$, which will be the signature of m_t.
We alter step **AL 1** and **2** as:

> $\mathcal{B}$ reveals (m_t, r) to the set A_t.
> The set A_t of participants verify that b_t is $m_t \cdot r^{e^2}$.

All other steps will remain the same, except that the signature of m_t is $m_t^d \cdot r$. The problem concerning this version is that it allows $\mathcal{B}$ to cheat. That is, $\mathcal{B}$, can propose one motion and choose an r, yet later when the motion is to be revealed, reveal a different motion, and a r'.

7. SUMMARY AND OPEN PROBLEMS

We have provided a verifiable democracy signature protocol. It allows a body $P_{i_1}, \ldots, P_{i_k}$ to transfer signature power to another body $P'_1, \ldots, P'_{n_t}$, such that we have transferred from a k out of n threshold scheme to a k_t out of n_t threshold scheme, yet, $P_1, \ldots, P_n$ do not give up signature power for future messages. Further, the protocol allows the set $P'_1, \ldots, P'_{n_t}$ to verify that this transfer has occurred without forcing anyone to form the signature. This protocol was outlined under the assumption that $P_{i_1}, \ldots, P_{i_k}$ was a subset of $P'_1, \ldots, P'_{n_t}$. This was a practical assumption allowing the protocol to be used at any time during a legislative session. (For example the protocol may be used many times in a given day). Although the paper was motivated for use in sessions where a majority vote will sign the law, nowhere in this protocol is there a requirement that $k_t = \lfloor \frac{n_t}{2} \rfloor + 1$. (For example, in some democratic

bodies, the threshold is a two-thirds.) The only requirement is that the number of potential cheaters is less than the majority. We have implicitly assumed that the number of cheaters is less than or equal to $n_t - k_t$. This can be achieved whenever $k_t \geq \lfloor \frac{n_t}{2} \rfloor + 1$. In fact, the protocol is secure for k_t which is not a function of n_t

Observe that a solution to the "dual problem" (creating a democratic threshold encryption scheme) is trivial. An obvious generalization would be to allow $P_{i_1}, \ldots, P_{i_k}$ to have members that are not contained in the set $P'_1, \ldots, P'_{n_t}$. The application would have to change slightly, presentation of motions must be made in front of the board $P_{i_1}, \ldots, P_{i_k}$, so that they can send partial signatures to $P'_1, \ldots, P'_{n_t}$.

Is it possible to modify the protocol so that it is robust? The answer is yes, identify the cheaters, remove them and run the protocol again.

Is it possible to achieve true democracy? That is, because we use a trusted dealer D, we do not achieve true democracy, since D possesses the power to sign all messages. Results by [2] cannot be used here because we are using safe primes. Thus, a protocol to achieve verifiable democracy without using a trusted dealer remains an open problem. Some other questions. Is it possible to achieve a solution to verifiable democracy that will satisfy all aspects of our model, and not allow the "motion sponsor" $\mathcal{B}$ to cheat? The ability to go from a k out of n threshold scheme to a k_t out of n_t threshold scheme is achieved by redistribution. Due to the nature of the problem, $k_t \leq k$. However many democratic bodies grow. For example, the United Nations has gone through an incredible amount of growth in the last ten years. Is it possible to create a democratic threshold scheme for which k_t may exceed k and then shrink again? Another question can a democratic threshold scheme be designed for other sharing schemes like DSS?

References

[1] G. Blakly. "Safeguarding cryptographic keys. " In *Proc. Nat. Computer Conf. AFPIPS Conf. Proc., 48.* pages 313-317, 1979.

[2] D. Boneh and M. Franklin. "Efficient generators of shared RSA keys" In *Advances of Cryptology - Crypto'97, LNCS 1294, Springer-Verlag,* pages 425-439, 1997.

[3] D. Chaum. "Zero-knowledge Undeniable Signatures" In *Advances of Cryptology - Eurocrypt'90, Springer-Verlag, LNCS 413,* pages 458-464.

[4] D. Chaum, J.H. Evertse, and J. van de Graff. "An improved protocol for demonstrating possession of discrete logarithms and some generalizations " In *Advances of Cryptology-Eurocrypt'87, LNCS 304, Springer-Verlag*, pages 127-141, 1988.

[5] Y. Desmedt and Y. Frankel. "Homomorphic zero-knowledge threshold schemes over any finite Abelian group " *SIAM J. on Discrete Math., vol.7, no. 4* pages 667-679, 1994.

[6] Y. Desmedt and S. Jajodia. Redistributing secret shares to new access structures and its applications. Tech. Report ISSE-TR-97-01, George Mason University, July 1997. ftp://isse.gmu.edu/pub/techrep/97_01_jajodia.ps.gz.

[7] Y. Frankel and Y. Desmedt. "Classification of ideal homorphic threshold schemes over finite Abelian groups, In *Advances in Cryptology- Eurocrypt'92, Springer-Verlag,* pages 25-34, 1992.

[8] Y. Frankel, P. Gemmel, P. MacKenzie, and M. Yung. "Optimal Resilience Proactive Public Threshold Cryptography " In 38^{th} *Annual Symp. on the Foundations of Computer Science, IEEE Computer Society Press*, 1997.

[9] E. Karnin, J. Greene, and M. Hellman. "On secret sharing systems. " *IEEE Tr. Inform. Theory, 29(1)*, pages 35-41, 1983.

[10] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. "Robust and efficient sharing of RSA functions" In *Advances in Cryptology - Crypto '96,. Lecture Notes in Computer Science 1109, Springer Verlag*, pages 157-172, 1996.

[11] R. Rivest, A. Shamir, and L. Adelman. "A method for obtaining digital signatures and public key cryptosystems. " *Commun. ACM, 21*, pages 120-126, 1978.

[12] A. Shamir. "How to share a secret " *Commun. ACM,* 22, pages 612-613, Nov., 1979.