Optimal round VSS with a non-interactive Dealer: VSS as a special case of VSR

Yvo Desmedt

The University of Texas at Dallas, USA

and

University College London,UK

May 31, 2016

©Yvo Desmedt

This is joint work with Kirill Morozov (Tokyo Institute of Technology, Japan).

The research in 2009-2010 on VSS started while Yvo Desmedt and Kirill Morozov were working at RCIS/AIST, Japan, respectively part time and full time.

The problem was addressed during several visits to Kyushu University and meetings abroad.

OVERVIEW

1. Redistribution of secret shares: some background
2. Our goals
3. Some preliminaries
4. Some definitions
5. Our VRS/VSS protocol
6. The types of errors
7. A note about the errors
8. The algebra behind the protocol
9. Our decoder: introduction
10. Our decoder
11. Open problem
12. Conclusions

1. REDISTRIBUTION OF SECRET SHARES

Many groups independently considered the following problem.

Suppose that participants in $\mathcal{P}$ are moving to a different organization, retiring, dying, etc. Then a new set of participants $\mathcal{P}'$ should receive shares. Unfortunately, the dealer is no longer available.

There are two approaches:

Trivial one: authorized participants, specified by $\Gamma_{\mathcal{P}}$, recompute the secret s, and then they play dealer and give new shares to parties in $\mathcal{P}'$ such that these authorized, as specified by $\Gamma_{\mathcal{P}'}$ can recompute the secret s.

Private approach: similar as before, but without the recomputation of the secret s.

Simmons posed this as an open problem (early 1990's). In Chen-Gollmann-Mitchell solution each recomputation grows the size of the shares. Desmedt-Jajodia avoided this growth. Others considered special cases, such as $\mathcal{P}' = \mathcal{P}$ and $\mathcal{P} \subset \mathcal{P}'$.

Both Chen-Gollmann-Mitchell and Desmedt-Jajodia only considered passive adversaries, as observed by Wong-Wang-Wing. Wong-Wang-Wing considered active adversaries in $\mathcal{P}$, but assumed all participants in $\mathcal{P}'$ to be honest! Moreover, their security is conditional. They called their protocol Verifiable Secret Redistribution (VRS). We will later see how VSS can be considered as a special case of VRS.

2. OUR GOALS

- Our original goal was to remove the interaction with the dealer in VSS.

  Removing this interaction has many advantages. We give some examples when the original data originates:

  – from a busy leader

  – when storing data before a flight

  – when the dealer used a pre-VSS area SS scheme

  – when the dealer had an accident after the dealing

  – when the dealer has limited resources, such as using a smartphone with a poor connection.

Despite many security experts warning against the use of cloud for storage, in our modern society everybody wants their data stored this way. “Multi-cloud Storage Toolkit” has been implemented by IBM (2010). Note that non-US cloud serves exist.

- Other questions that we raised was whether we need as much randomness as most VSS schemes use. Most VSS schemes that have few rounds require the dealer to have $O(t^2)$ random values as large as the secret.

A trivial approach to remove the interaction with the dealer is parties execute a secure multi-party computation. (We recently learned Cramer et al. also observed this). However, this increases the round complexity, which by itself was a major research problem 5-6 years

ago. When analyzing the round complexity of VSS, one assumes that broadcast is free (i.e., does not require extra interaction).

So, a natural question became whether we can:

- achieve all above while having 3 rounds for both VRS and VSS.

3. SOME PRELIMINARIES

As observed by McEliece-Sarwate, when we let $k = t + 1$ and $\mathbf{u} = (s, r_1, r_2, \ldots, r_t) \in F^{t+1}$, where s is the secret and r_i are uniformly random, the shares s_j $(1 \leq j \leq n)$ the n parties receive in Shamir's secret sharing scheme, can be regarded as a codeword $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, generated by a $k \times n$ generator matrix G, as follows. G corresponds to the generator matrix of the Generalized Reed-Solomon code and

$$\mathbf{s} = \mathbf{u} \cdot G,$$

where the j-th column in G corresponds to $(1, \alpha_j, \alpha_j^2, \ldots, \alpha_j^t)$, and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are distinct non-zero elements of a finite field F.

According to the above, we denote the generator matrix of the

$t+1$-out-of-n Shamir secret sharing scheme by G, and we use H to indicate the corresponding parity check matrix.

(Note that we can generalize the use of G and H to cover linear secret sharing schemes for general access structures.)

4. SOME DEFINITIONS

We assume that the protocol is synchronous.

We assume that the number of dishonest parties in $\mathcal{P}$ are t and that $n \geq 3t + 1$ and that the number of dishonest parties in $\mathcal{P}'$ are t' and that $n' \geq 3t' + 1$. In the case of VSS $t' = t$, $n' = n$ and $\mathcal{P}' = \mathcal{P}$. For simplicity, we assume for the VRS case that $\mathcal{P} \cap \mathcal{P}' = \emptyset$, but we allow any dishonest t parties in $\mathcal{P}$ to collaborate with any dishonest t' parties in $\mathcal{P}'$.

(Note that we can generalize this to access structures $\Gamma_{\mathcal{P}}$ and $\Gamma_{\mathcal{P}'}$ and their respective adversary structures, provided the Q^3 condition is satisfied over $\mathcal{P}$ and $\mathcal{P}'$.)

We do not assume any restrictions on adversary's computational

power.

Definition 1. n values $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ are called almost consistent shares in an $t+1$-out-of-n Shamir secret sharing scheme in which $n \geq 3t+1$ when $\mathbf{s}$ is at Hamming distance at most t from a codeword formed using $\mathbf{u} \cdot G$, where G is the Generator Matrix.

(Note that we can generalize this to consider an error caused by a subset of participants in the adversary structure).

5. Our VRS/VSS protocol

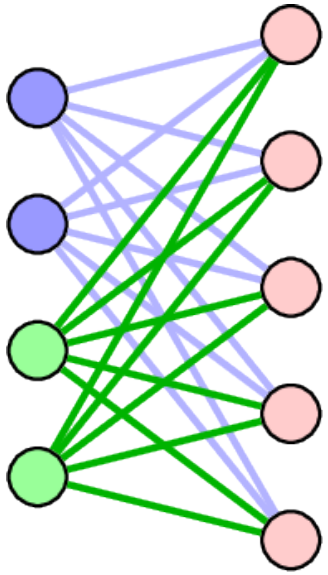
In sharp contrast with the published literature, the dealer uses ordinary Shamir secret sharing and we do not use any extra randomness. (In general we assume a linear secret sharing, in which any honest subset can recover all randomness.)

If the dealer is an external party, the dealer will stop participating.

In VSS, we need to check that the parties in $\mathcal{P}$ received "consistent" shares of s from the dealer. In VRS, we need to check that the parties in $\mathcal{P}'$ received "consistent" shares from $\mathcal{P}$.

An important part in both our VRS/VSS protocols is that parties redistribute their shares, in a way very different from: Desmedt-Jajodia and very different from Ben Or-Goldwasser-

Wigderson and Cramer-Damgård-Maurer.



The nodes are regarded as participants. We regard that all participants in $\mathcal{P}$ are on the left, and these in $\mathcal{P}'$ are on the right. The edges will correspond to private communications.

Parties in $\mathcal{P}$ and in $\mathcal{P}'$ can behave dishonestly, which we explain further. Parties in $\mathcal{P}$ doing this are denoted by $\mathcal{J}$ and similarly we have $\mathcal{J}'$. (Note that if $\mathcal{P}' = \mathcal{P}'$, we do not necessarily have that

$\mathcal{J}' = \mathcal{J}$.)

Step 1 All parties in $\mathcal{P}$ make shares of their shares, as follows. Each party $P_j \in \mathcal{P}$ chooses t' uniformly random values $r_{i,j}$ $(1 \leq i \leq t')$ to form $\mathbf{u}_j^T = (s_j, r_{1,j}, r_{2,j}, \ldots, r_{t',j})^T$, and computes

$$\mathbf{s}_j^T = (s_{1,j}, s_{2,j}, \ldots, s_{n',j})^T = G'^T \cdot \mathbf{u}_j^T$$

and sends $s_{i,j}$ to $P'_i \in \mathcal{P}'$ privately.

Step 2 Each party $P'_i \in \mathcal{P}'$ having received $\mathbf{s}'_i = (s'_{i,1}, s'_{i,2}, \ldots, s'_{i,n})$ from $P_j \in \mathcal{P}$ $(1 \leq j \leq n)$ computes

$$\mathbf{temp}_i = \mathbf{s}'_i \cdot H^T$$

and broadcasts the $n - k$ values in $\mathbf{temp}_i$ to all parties in $\mathcal{P}'$.

We let $Temp$ be the $n' \times n - k$ matrix in which its i-th row is $\mathbf{temp}_i$.

Step 3 Each party $P'_i \in \mathcal{P}'$ runs a non-interactive decoding process (see further), which will identify some appropriate (see further) $\mathcal{J}$ and $\mathcal{J}'$.

Based on the results from above decoding, honest parties in $\mathcal{P}'$ conclude the original dealer was dishonest or not. If declared honest, they correct, without interaction:

- in the VRS case: their shares-of-shares obtained, and then apply the Desmedt-Jajodia compression.
- in the VSS case: their original shares obtained from the dealer.

6. THE TYPES OF ERRORS

Let $S = [s_{i,j}]$, the $n' \times n$ secret matrix of shares-of-shares. We first identify the types of malicious errors (focus: threshold case).

Type i) The first element of $\mathbf{u}_j^T$ must be a consistent share s_j, which we call the share-valid condition (see also Wong-Wang-Wing).

There are two ways that this condition could be violated:

- the dealer gave some parties inconsistent shares
- some party (or parties) P_j replaces s_j with some randomness when performing the redistribution protocol.

We regard both as an error in the codeword $\mathbf{s}$ at location j, we call f_j the corresponding error, which defines

$\mathbf{f} = (f_1, f_2, \ldots, f_n)$.

Remark: if the dealer made more than t such errors, the dealer will eventually be declared dishonest (see further).

Type ii) The shares-of-shares $s_{i,j}$ must be consistent, i.e., for each fixed j, $(\alpha'_i, s_{i,j})$ must correspond to points on a polynomial of degree at most t'. If the shares-of-shares are non-consistent, then $\mathbf{s}_j^T = G'^T \cdot \mathbf{u}_j^T$ is replaced by P_j into

$$\mathbf{s}_j^T + \mathbf{e}_j^T, \qquad \text{where} \quad \mathbf{e}_j^T \text{ is an } n'\text{-column.}$$

To describe the impact of these inconsistent shares caused by all dishonest parties in $\mathcal{P}$, we introduce an $n' \times n$ matrix E, where the j-th column of E is only nonzero when $P_j \in \mathcal{J}$ and then this j-th column is $\mathbf{e}_j^T$.

Type iii) Wong-Wang-Wing assumed parties in $\mathcal{P}'$ to be honest. We do not.

Up to t' parties $P_i' \in \mathcal{P}'$ can each broadcast their incorrect values for $\mathbf{temp}_i$, which we denote as having them broadcast

$$\mathbf{temp}_i + \mathbf{e}_i'.$$

To describe the impact of all dishonest parties in $\mathcal{P}'$, we introduce an $n' \times n - k$ matrix E', where the i-th row of E' is only non-zero when $P_i \in \mathcal{J}'$ and such a row corresponds to $\mathbf{e}_i'$.

7. A NOTE ABOUT THE ERRORS

In VRS the participants in $\mathcal{P}$ are the distributed equivalence of the role of the dealer in VSS.

In VSS we cannot distinguish between the following two cases:

Case 1 All participants are honest, but the dealer gives t parties inconsistent shares.

Case 2 The dealer is honest, but at most t participants pretend having received incorrect shares of the dealer.

The equivalence in the case of VRS is:

Case a All participants in $\mathcal{P}'$ are honest, but t' of the participants in $\mathcal{P}'$ receive incorrect shares-of-shares.

Case b All parties in $\mathcal{P}$ are honest, but at most t' participants in $\mathcal{P}'$ pretend having received incorrect shares.

This implies that for some type of errors, we will not be able to uniquely identify $\mathcal{J}$ and $\mathcal{J}'$.

Note that we are not interested in finding who caused these errors! We are interested in making certain that honest parties in $\mathcal{P}'$ receive correct shares, and in the VSS case, come to correct shares for honest parties in $\mathcal{P}$ or declare the dealer dishonest.

8. THE ALGEBRA BEHIND THE PROTOCOL

Lemma 1. *When the parties in $\mathcal{P}$ gave consistent shares, but the share-valid condition has been violated, each column in $Temp$ are almost consistent shares of the $n-k$ syndromes corresponding to $\mathbf{f} \cdot H^T$.*

Proof: Since S is replaced by $S+E$, we have $Temp = (S+E) \cdot H^T + E'$. Now, $S = G'^T \cdot U$, where U is a $k' \times n$ matrix in which the first row is $\mathbf{s}+\mathbf{f}$. So, using a block matrix $U = [\mathbf{s}+\mathbf{f} \mid R]^T$, where R is a $t' \times n$ matrix, or $U = [\mathbf{u} \cdot G + \mathbf{f} \mid R]^T$. This gives $Temp = G'^T \cdot [\mathbf{u} \cdot G \cdot H^T + \mathbf{f} \cdot H^T \mid R \cdot H^T]^T + E \cdot H^T + E'$. Using the fact $G \cdot H^T = 0$, where 0 is the $k \times n-k$ zero matrix, this gives us:

$$Temp = G'^T \cdot \begin{bmatrix} \mathbf{f} \cdot H^T \\ R \cdot H^T \end{bmatrix} + E \cdot H^T + E' \quad (1)$$

Since $\mathcal{P}$ gave consistent shares, $E = 0$. Also, $\mathbf{f} \cdot H^T$ are the syndromes caused by having a violation of the share-valid condition. Since R is uniformly random, and H of full rank, $R \cdot H^T$ is a $t' \times n - k$ random matrix, which guarantees that the multiplication on the left by G'^T in Eqn. 11 makes the result shares of the $n - k$ syndromes corresponding to $\mathbf{f} \cdot H^T$. The fact that for each of these $n - k$ syndromes the n' values are almost consistent shares follows from the fact that E' has at most t' non-zero rows. □

Corollary 1. *$Temp$ does not leak anything about s, the original secret.*

Proof: From Eqn. 11, it follows that $Temp$ is independent of the secret s. □

9. OUR DECODER: INTRODUCTION

In Lemma 1 we assumed that the parties in $\mathcal{P}$ gave consistent shares. How can we remove this assumption?

A problem we may encounter is that some parties in $\mathcal{P}$ may give very inconsistent shares, poisoning the protocol.

Solution: we want to remove the poison! Problem: since we want constant rounds, we can not go back and ask to recompute shares of the syndromes ignoring some inputs.

So, what saves us?

1. a parity check matrix H is not unique. Any invertible linear combinations of the $n-k$ rows of H form a new parity check matrix.

2. a parity check matrix can be put in systematic form. This means that we get $H = [-R'_B \; I_{n-k}]$, where $B \in \Gamma_{\mathcal{P}}$.
   Note now that for *some* syndrome $n-k-1$ entries of the received word will not be used, since the corresponding column in H will have $n-k-1$ zero entries. Similarly, for *two* syndromes $n-k-2$ entries of the received will not be used, etc.

Our decoder exploits the following properties to find the Type (iii) errors. We now prove the mathematics behind this idea.

Corollary 2. *If the Q^3 property is satisfied, for every two maximal sets $A_1, A_2 \in \Lambda_{\mathcal{P}}$, we can write $H = [R''_B \; V_{n-k}] \cdot F_{\pi_B}$ where*

$V_{n-k} \in \mathbb{F}^{n-k \times n-k}$ *is an invertible matrix and* F_{π_B} *is a permutation matrix*

Proof: For the threshold case, take $B = \mathcal{P} \setminus (A_1 \cup A_2)$.
(Generalized: skipped). □

Corollary 3. *(Syndrome Input Exclusion Corollary) If the* Q^3 *property is satisfied, for every set* $A \in \Lambda_{\mathcal{P}}$*, when taking some appropriate linear combinations of the syndromes, for some syndromes, the errors caused by* $A \in \Lambda_{\mathcal{P}}$ *will be excluded and the corresponding linear combination(s) will be zero.*

Proof: We use the notations used in the proof of Corollary 2. By

multiplying the syndromes with V_{n-k}^{-1}, we obtain that: $\mathbf{e}' I_{n-k}$, where $\mathbf{e}'$ is caused by two unauthorized sets. Therefore, the columns in I_{n-k} that are orthogonal on $\mathbf{e}'$ will give syndromes equal to zero. Since rows in I_{n-k} have only a single non-zero entry, and since in this corollary, we consider errors caused by a *single unauthorized* set, we obtain the claim. □

10. OUR DECODER

All parties in $\mathcal{P}'$ will run in Step 3 of our protocol on their own, i.e., without any interaction.

1. Loop over all possible dishonest sets $A \in \Lambda_{\mathcal{P}}$:
   i. Loop over all possible sets $B \subset \mathcal{P} \setminus A$ and compute compute $TempV_B^{-1}$, where $V_B^{-1} \in \mathbb{F}_2^{n-k \times n-k}$ is an invertible matrix as specified in Corollary 2 and the Syndrome Decoding Input Exclusion Corollary, and where $Temp$ is the matrix of n' shares of the $n-k$ syndromes.
   Due to the fact that V_B^{-1} forces an identity matrix in $V_B^{-1}H$ (spread over columns), we can split the syndromes into two categories, these for which we have (almost) consistent shares, and these we do not. If for the last loop, we get that the locations of the

inconsistent shares are caused by the same A, we have identified A, else we try another one.

For the VSS case: If both loops do not terminate prematurely, the dealer is declared dishonest.

2. When we found A, we only consider the linear combinations that gave us almost consistent shares. Each party computes from these remaining almost consistent combined shares of the j-th syndrome, the consistent shares and then the actual syndromes using the reconstruction protocol of the secret sharing scheme (e.g., Lagrange).

3. For the VSS case: If the remaining linearly combined syndromes are zero, then the protocol succeeds,

else corrects the shares s_i, i.e. compute the error vector $\mathbf{f} \in \mathbb{F}^n$ using the syndrome decoding;

if the errors in $\mathbf{f}$ with the union of A span the set not in $\Lambda_{\mathcal{P}}$ or if the error correction fails, then declare D dishonest,

else each involved party P_i accepts the corrected share $\widetilde{s}_i = s_i - f_i$ and the protocol succeeds.

For the VRS case: Having found "the" dishonest parties $\mathcal{J}$ in $\mathcal{P}$ and "the" dishonest parties $\mathcal{J}'$ in $\mathcal{P}'$, the parties can ignore the shares of shares received from the parties in $\mathcal{J}$. They then use the Desmedt-Jajodia compression using any honest subset of $\mathcal{P} \setminus \mathcal{J}$.

11. OPEN PROBLEM

In the general adversary structure case, our decoder is efficient.

However, that is not true for the threshold case. So, the open problem is how to make an efficient decoder.

Do we have a suggestion?

11. OPEN PROBLEM

In the general adversary structure case, our decoder is efficient.

However, that is not true for the threshold case. So, the open problem is how to make an efficient decoder.

Do we have a suggestion?

Clever monkeys are just copycats (2012 study!!)

Since we copy, we introduce more syndromes. So, we define:

$$Syn = H' \cdot Temp,$$

giving an $n' - k' \times n - k$ matrix. As we learned:

$$Temp = G'^T \cdot \begin{bmatrix} \mathbf{f} \cdot H^T \\ R \cdot H^T \end{bmatrix} + E \cdot H^T + E'$$

Multiplying at the right with H' then gives:

$$Syn = H' \cdot Temp = H' \cdot E \cdot H^T + H' \cdot E' \qquad (2)$$

which becomes independent of $\mathbf{f}$. We then attempt to make a Peterson-Gorenstein-Zierler decoder. We first define error-locator polynomials corresponding to errors done by $\mathcal{J}$ and produced by

$\mathcal{J}'$, which we write respectively as:

$$\Lambda(x) = \prod_{P_j \in \mathcal{J}} (1 - \alpha_j x) \quad \text{and} \quad \Lambda'(y) = \prod_{P'_j \in \mathcal{J}'} (1 - \alpha_j y).$$

This then gives (proceeding as Peterson-Gorenstein-Zierler):

$$\sum_{i=0}^{\tau'} \lambda'_i \cdot \left(\sum_{j=0}^{\tau} Syn[\tau' + l' - i, \tau + l - j] \cdot \lambda_j \right) = 0 \quad (3)$$

where $\tau = |\mathcal{J}|$ and $\tau' = |\mathcal{J}'|$. Which in tensor notation becomes:

$$\lambda' \mathbf{Syn} \lambda = 0,$$

where λ and λ' are tensors of order 1 (vectors) and $\mathbf{Syn}$ is a tensor of order 4. Since both λ and λ are unknowns, we have a non-linear

set of equations. Above corresponds to a bilinear form of a quadratic form.

While in Peterson-Gorenstein-Zierler, the matrix notation allowed an efficient decoder finding the error locator, in our case, we do not know how to efficiently solve the above tensor equation. Moreover, we know that $(\mathcal{J}, \mathcal{J}')$ might not be unique.

12. CONCLUSIONS

Recent research on VSS has focused on rounds. We believe there are other aspects worth analyzing, such as randomness complexity, communication complexity, etc.