On alternative approach for verifiable secret sharing†

Kamil Kulesza[1], Zbigniew Kotulski[1], Josef Pieprzyk[2]

[1]Institute of Fundamental Technological Research, Polish Academy of Sciences, ul.Świętokrzyska 21, 00-049 Warsaw, Poland, e-mails: Kamil.Kulesza@ippt.gov.pl, Zbigniew.Kotulski@ippt.gov.pl
[2]Department of Computing, Macquarie University, NSW2109, Australia, e-mail: josef@ics.mq.edu.au

Abstract.
The proposed approach works for any underlying secret sharing scheme. It is based on the concept of verification sets of participants, related to authorized set of participants.
The participants interact (no third party involved) in order to check validity of their shares before they are pooled for secret recovery. Verification efficiency does not depend on the number of faulty participants.

1. Introduction

Everybody knows situations, where permission to trigger certain action requires approval of several selected entities. Equally important is that any other set of entities cannot trigger the action. Secret sharing allows a secret to be split into different pieces, called shares, which are given to the participants, such that only certain groups (authorized sets of participants) can recover the secret. Secret sharing schemes (SSS) were independently invented by George Blakley [1] and Adi Shamir [2]. Many schemes have been presented since, for instance, Asmuth and Bloom [3], Brickell [4], Karin-Greene-Hellman (KGH) [5].
Once secret sharing was introduced, it was found that it can be easily compromised by misbehaving parties. Hence, the ability to perform secret consistency verification and detection of cheaters (e.g., [6]) is very important. One of solutions is to use Verifiable Secret Sharing (VSS), for instance see [7]. It can be done for the conditionally secure secret sharing (e.g.[7]), but also for the unconditional secure secret sharing, for instance see [8]. The verification capacity usually comes at a price. This fact is related to the paradox stated by David Chaum, that no system can simultaneously provide privacy and integrity.
One interesting proposal, called Robust Sharing of Secrets, was presented by Tal Rabin in [9]. In fact it can be considered, as the special case of our construction.
We propose an approach to verification that can be seen as a distributed computation of authentication codes. The participants interact, without cooperation of the third party, in order to check validity of their shares before they are pooled for recovery of the secret. Our proposal can support the recovery of corrupted shares, but in order to keep the presentation simple, we do not address this in the paper.
There are two main pillars of the construction:

a. defining sets of participants that interact
b. defining the verification function

The outline of the paper is the following: in Section 2, definitions and notation for an alternative approach are provided. The next section brings discussion on verification operations and control functions. This leads to the verification protocol description in the Section 4. The last section contains concluding remarks.

2. Preliminaries

2.1 Definitions

It is assumed that the shares are assigned to the participants by the honest dealer.

† The paper is extension of results presented at ESORICS2002 in Zurich.

We propose interactive verification protocol (VP) that utilizes the concept of a verification set of participants. A verification set of participants (VSoP) is the set of the secret participants that are needed to verify their shares. In order to test the validity of the shares, all shares belonging to the participants from the VSoP are required.
Verification structure (Γ_V) is the superset containing all verification sets of shares. Both terms (VSoP, Γ_V) are closely related to the authorized set of participants and general access structure (Γ), respectively. In the general case, one can investigate all possible combinations of relations between Γ_V and Γ. In order to simplify the presentation, we restrict the discussion to verification sets of participants that are subsets of authorized sets of participants. In this case, before the secret is recovered, all shares needed to recover the secret are thoroughly tested in the smaller sets.

2.2 Verification protocol

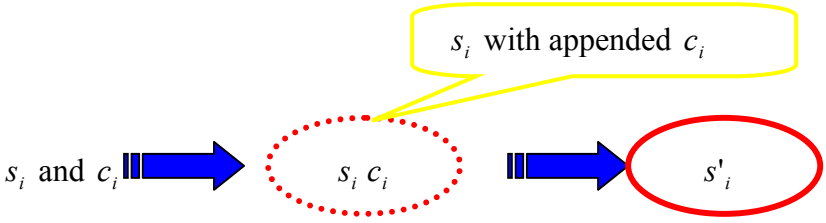
We present a simplified description of general verification protocol. Full treatment would require more formal approach, for instance using methods for statistical hypotheses testing. For each VSoP there are two possible outcomes of verification protocol (VP):
a. **negative verification.** The result means that at least one share in VSoP is invalid with probability p_1**, or**
b. **positive verification**. The result means that all shares in VSoP are valid with probability p_2.
At least one of p_1, p_2 has to be equal 1. In this paper we discuss case where $p_1 = 1$, while $p_2 \leq 1$.

Example 1

Consider (v,t,n) threshold secret sharing schemes, where v denotes the number of participants in verification sets. When participants belonging to the authorized set want to recover the secret, they first run VP for all VSoP-es contained in that set. For $p_1 = 1$ and $p_2 \leq 1$, the probability of positive verification for invalid share(s) is not bigger than $(1-p_2)^{t-1}$ ■

2.3 Notation

1. Take any secret sharing scheme (SSS) over general access structure, with the k participants $P_1, P_2, \ldots, P_k$ and corresponding secret shares $s_1, s_2, \ldots, s_k$. Let's denote C_0 as the combiner algorithm for that secret sharing scheme.
2. In order to implement VSS each secret share s_i should be extended by the control part c_i to form *extended secret share* s'_i.



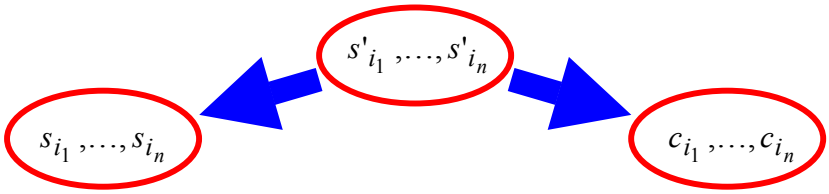
3. Let $C_1(a_1,...,a_\alpha), C_2(b_1,...,b_\beta)$ denote the combiner algorithms for two SSS-es operating on the sets of the shares $a_1,...,a_\alpha$ and $b_1,...,b_\beta$, respectively. These two SSS-es may have different access structures.

Remark: further in the text we denote p_2 as $P(s'_{i_1},...,s'_{i_n})$.

3. Building blocks

3.1 Operations in verification set of participants (VSoP)

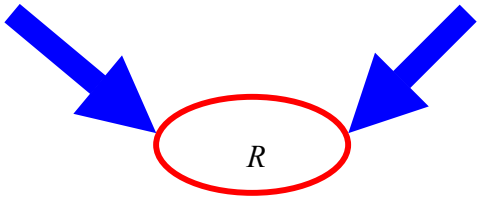
1.Let $s'_{i_1},\ldots,s'_{i_n}$ be the extended secret shares, such that each share belongs to some participant P_{i_α} ($\alpha = \{1,...,n\}$, $n \le k$) and set $\{P_{i_1},...,P_{i_n}\}$ forms VSoP.



2.Combine $s_{i_1},\ldots,s_{i_n}$ using C_1 to get R_s (resulting secret part). Formally $C_1(s_{i_1},...,s_{i_n}) = R_s$.

3.Combine $c_{i_1},\ldots,c_{i_n}$ using C_2 to get R_c (resulting control part). Formally $C_2(c_{i_1},...,c_{i_n}) = R_c$

4. R is the total result, equal to $R_s R_c$ (R_c appended to R_s) and set $\{P_{i_1},...,P_{i_n}\}$ forms VSoP.



Observation 1: when $s'_{i_1},\ldots,s'_{i_n}$ are valid, then $f(R_s) = R_c$ for every s'_i in the VSoP.

Observation 2: a different C_1, C_2 and $f(x)$ can be used for each VSoP.

Observation 3: $R_s R_c$ are dependent. For $|\Gamma_V| > 1$ and perfect SSS-es, individual s_i, c_i are independent. This is specially true if Γ allows random assignment of c_i to s_i by the Dealer. In order to focus on the major concepts we stop the topic at this point.

3.2 Construction of the control function $f(x)$

Description:

$f(x)$ takes an l-bit vector x and computes the m-bit image/control number.

Requirements:

a. VSoP perfectness. Any VSoP must not provide any information about the secret.
b. a faulty participant (a cheater) has the same probability of guessing the secret as the remaining (honest) participants.
c. $f(x)$ should be efficient to compute.

Sample candidates for $f(x)$***:***

a. for $m = 1$ use a balanced, nonlinear Boolean function (e.g., modified bent function).
b. for $m > 1$, one can use a vector of m different balanced, nonlinear Boolean functions. Consecutive values of functions from the vector are written as a binary sequence to form the m-bit control number, for instance see [10].
c. check-digit schemes, for instance one based on D_5 symmetry group, see [11].
d. hash functions

4. Verification Protocol (VP)

4.1 Protocol description

Verification protocol (one round). The participants can verify their shares without co-operation of a third party.

1. For any verification set of shares (VSoP) compute R *equal to* $R_s R_c$.

2. Compute $f(R_s)$.

3. Test the relationship between $f(R_s)$ *and* R_c *:*

if $f(R_s) \neq R_c$ *at least one of the shares in VSoP is invalid (verification is negative/ negative verification result)*

if $f(R_s) = R_c$ *all shares in VSoP are valid with some probability* $P(s'_{i_1},...,s'_{i_n})$. ■

The protocol described above is performed for all VSoP contained in the authorized set of participants, that want to recover the secret.

4.2 On probability $P(s'_{i_1},...,s'_{i_n})$

Let C_1 , C_2 be the combiner algorithms for perfect secret sharing schemes. In addition the impact on R resulting from any change of bit(s) in $s'_{i_1},\ldots,s'_{i_n}$ cannot be predicted in at least one of C_1 , C_2 .

$P(s'_{i_1},...,s'_{i_n})$ depends on the R_c length (m-bits) , we think that, for properly chosen $f(x)$, it is related to the probability of guessing m-bits number. For the given m there are $\left(\frac{1}{2}\right)^m$ m-bits numbers, hence $P(s'_{i_1},...,s'_{i_n}) = 1 - \left(\frac{1}{2}\right)^m$ for properly chosen $f(x)$.

4.3 Example 3: (v,t,n) secret sharing scheme

We assume that $f(x)$ is balanced, nonlinear Boolean function with $P(s'_{i_1}, s'_{i_2}) = \frac{1}{2}$

x	$f(x)$
⋮	⋮
00010	0
⋮	⋮
01111	1
10000	1
⋮	⋮

10010	1
⋮	⋮
11101	0
⋮	⋮
11111	0

Take a $(3,4)$ threshold secret sharing where the secret was shared using the Shamir method (C_0 is Shamir combiner algorithm).

Participants P_1, P_2, P_3, P_4 hold secret shares s'_1, s'_2, s'_3, s'_4 respectively.

Authorized sets of participants: $\{P_1, P_2, P_3\}, \{P_1, P_2, P_4\}, \{P_1, P_3, P_4\}, \{P_2, P_3, P_4\}$

Verification sets of participants: $\{P_1, P_2\}$, $\{P_1, P_3\}$, $\{P_1, P_4\}$, $\{P_2, P_3\}$, $\{P_2, P_4\}$, $\{P_3, P_4\}$

Let $g(x) = 7 + 5x + 3x^2$ be random polynomial over $GF(31)$.

$x_i = i$ for $i = 1,2,3,4 \quad i \in \{1,2,3,4\}$

$s_1 = g(1) = 15 = 01111_2$, $c_1 = 1$ resulting in $s'_1 = 011111$

$s_2 = g(2) = 29 = 11101_2$, $c_2 = 0$ resulting in $s'_2 = 111010$

$s_3 = g(3) = 18 = 10010_2$, $c_3 = 1$ resulting in $s'_3 = 100101$

$s_4 = g(4) = 13 = 01101_2$, $c_4 = 1$ resulting in $s'_4 = 011011$

Let both of C_1, C_2 be combiner algorithm for KGH secret shares scheme.

Now consider authorized set $\{P_1, P_2, P_3\}$

Such an authorized set has the following verification sets: $\{P_1, P_2\}$, $\{P_1, P_3\}$, $\{P_2, P_3\}$.

For:

$\{P_1, P_2\}$ $R_s = s_1 \oplus s_2 = 10010$ and $R_c = c_1 \oplus c_2 = 1$,

$\{P_1, P_3\}$ $R_s = s_1 \oplus s_3 = 11101$ and $R_c = c_1 \oplus c_3 = 0$,

$\{P_2, P_3\}$ $R_s = s_2 \oplus s_3 = 01111$ and $R_c = c_2 \oplus c_3 = 1$

Verification protocol

1st round for $\{P_1, P_2\}$ $f(R_s) = f(10010) = 1 = R_c$, hence s'_1, s'_2 are valid with $P(s'_1, s'_2) = \frac{1}{2}$

2nd round for $\{P_1, P_3\}$ $f(R_s) = f(11101) = 0 = R_c$, hence s'_1, s'_3 are valid with $P(s'_1, s'_3) = \frac{1}{2}$

3rd round for $\{P_2, P_3\}$ $f(R_s) = f(01111) = 1 = R_c$, hence s'_2, s'_3 are valid with $P(s'_2, s'_3) = \frac{1}{2}$

■

Discussion of VP results:

1. No negative verification result was obtained in all rounds of VP.
2. Each of s'_i is valid with probability $P = 1 - \left(\frac{1}{2}\right)^2 = 0{,}75$

5. Concluding remarks

The presented VSS has the following features:

a. it works for any secret sharing scheme,
b. it does not require cooperation of the trusted third party,
c. it can be implemented for a general access structure,

d. its efficiency is not related to the number of dishonest participants in a properly chosen frame,
e. it does not weaken the security parameter of the underlying secret sharing scheme.

The last requirement means that no extra information about the secret is revealed.
For example a perfect secret sharing scheme, when used with proposed VSS, still remains perfect. The information rate for the secret shares is always smaller than one, even for the underlying ideal secret sharing schemes. In the chosen design it can be made close to one.
Apart from the scope of this paper, authors are investigating the following issues:

a. description of the proposed approach as an authentication code,
b. using VSoP-es that are not subsets of the authorized set,
c. a mechanism for recovery of corrupted shares.

Acknowledgement

1. Authors want to thank Dr. Martin Hirt, from ETH Zurich, for discussion and comments.
2. The paper was finished during visit to Rhodes University, Grahamstown, South Africa. Special thanks to Mrs. Caro Watkins for reading and checking the manuscript.

7. References

[1] Blakley G.R. 1979. 'Safeguarding cryptographic keys'. *Proceedings AFIPS 1979 National Computer Conference*, pp. 313-317.
[2] Shamir A. 1979. 'How to share a secret'. *Communication of the ACM* 22, pp. 612-613.
[3] Asmuth C. and Bloom J. 1983. 'A modular approach to key safeguarding'. *IEEE Transactions on Information Theory* IT-29, pp. 208-211
[4] Brickell E.F. 1989. 'Some ideal secret sharing schemes' *Journal of Combinatorial Mathematics and Combinatorial Computing* 6, pp. 105-113.
[5] Karnin E.D., J.W. Greene, and Hellman M.E. 1983. 'On secret sharing systems'. *IEEE Transactions on Information Theory* IT-29, pp. 35-41.
[6] Tompa M., Woll H. 1988. 'How to share a secret with cheaters'. Journal of Cryptology, pp. 133-138.
[7] Stadler M. 1997. 'Publicly verifiable secret sharing'. *Lecture notes in Computer Science,* pp.190-199 *(Advances in Cryptology – EUROCRYPT'96)*
[8] Chaum D., Crepeau C. and Damgard I. 1988. 'Multiparty unconditionally secure protocols'. *Proc. 20th Annual Symp. on Theory of Computing*, ACM, pp. 11-19.
[9] Rabin T. 1994. 'Robust Sharing of Secrets when the Dealer Is Honest or Cheating'. *Journal of ACM*, Vol. 41, No. 6, pp.1089-1109
[10] Pieprzyk J., Xian-Mo Zhang 2001. 'Nonlinear secret sharing immune against cheating'. *The 2001 International Workshop on Cryptology and Network Security, Workshop Proceedings of the Seventh International Conference on Distributed Multimedia Systems, September 26-28, Tamkang University, Taipei, Taiwan*, pp. 154-161
[11] Gumm H.P. 1985. 'A New Class of Check-Digit Methods for Arbitrary Number Systems'. *IEEE Transactions on Information Theory* IT-31, pp. 102-105