DOI 10.34658/jacs.2018.26.2.57-72

JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 26 No. 2 (2018), pp. 57-72

Numerical Models of Hierarchical Threshold Secret Sharing and Broadcasting with Encryption

Joanna Kapusta[1], Ryszard Smarzewski[2]

[1]*The John Paul II Catholic University of Lublin*
Institute of Mathematics and Computer Science
ul. Konstantynow 1H, 20-708 Lublin, Poland
jkapusta@kul.pl

[2] *The John Paul II Catholic University of Lublin*
Institute of Mathematics and Computer Science
ul. Konstantynow 1H, 20-708 Lublin, Poland
rsmax@kul.pl

Abstract. *In the paper there are presented two new models of encrypted hierarchical secret sharing schemes based on barycentric Hermite formula. Moreover an application of the second scheme to design a novel broadcast encryption protocol is proposed. The protocol allows to send a decoding key to any user via broadcast channel and revoke some users without the necessity of updating encrypted private keys of the other users of the system. To ensure the safety of user private keys the protocol uses one-way functions that fulfill special conditions.*

Keywords: *hierarchical secret sharing, barycentric Hermite weights, broadcast, encryption.*

1. Introduction

Broadcast transmission consists in sending a signal to multiple recipients without indicating the address. Such a signal may be encrypted to ensure access for a specific group of users - then only users who know the decryption method have access to the sent information. Such a data transfer method is used, among others, in the case of pay TV and sharing the data via the Internet, where only those users who have fulfill certain conditions (for example, made the charges) have access to the data. Recipients may dynamically change during the time period: new users who have made the payment can join and users who have not payed may lose their rights. Thus it is important to exchange decoding key called the session key, regularly. This requires constructing key exchange protocol, which allows safe and effective dissemination of new session key among the entitled users.

Let *GC* be a group controller i.e. the unit, which is responsible for providing keys to authorized users, and broadcasting messages. The activities carried out by *GC* can be divided into two parts - system initialization and the broadcasting.

During initialization *GC* prepares private key for each user of the system, then *GC* activities are repeated in cycles. A single cycle starts with establishing the users who in the next cycle will be revoked, i.e. they lose access to encrypted information. Then *GC* generates a new session key and broadcasts it in specially-prepared message, which allows obtain the key only entitled users. After reconstituting the key, users have access to information until the beginning of a new cycle and establishing a new session key. Details of the particular stages of broadcast communication are shown in the following scheme.

A scheme of a broadcast communication:

1. System initialization:

    i) Determining system parameters, indicating *GC*, choosing cryptographic algorithms.

    ii) Registering users - the user receives a new private key K_U, which allows him to access the system.

2. Broadcast part:

    i) Broadcast key - *GC* sends an enabling block T, in which the session key S is hidden (only authorized users can recover it).

ii) Signal transmission - GC encrypts the message M using the session key S and encryption algorithm E, and then sends the received ciphertext $C = E(S, M)$.

iii) Signal reconstruction - the user, using his private key K_U obtains the session key S from the enabling block T, then using S decrypts the message by reconstructing M from C.

Over the years, a number of broadcast protocols have been proposed. They implemented various solutions to revoke and restore users dynamically and to detect illegal system access attempts [3], [5], [8], [10], [11]. In particular, M. Naor and B. Pincas [10] proposed a model using the secret sharing scheme in which the key S is free polynomial coefficient of $p(x)$ given in power base representation and the reconstruction S is being done using Lagrange interpolation formula. Modification of this model consisting in a change of the key selecting way and its reconstruction is given in [7]. In a modified model version the key S is a leading polynomial coefficient and its reconstruction is based on Newton's interpolation formula. Some of the of papers (e.g. [21]) presenting algorithms for identifying illegal users and tracing traitor (whose keys were used to create keys for illegal users). In these algorithms it is assumed that keys of illegal users are linear combinations of keys of dishonest users.

The mentioned models use the idea proposed in [4] consisting in transferring the secret Shamir scheme to the exponent. The safety of these models is based on decision Diffie-Helman problem [1].

2. Secure hierarchical secret scharing

The first secret sharing algorithm of dividing an information between members of a fixed participant group was proposed by Shamir [15] in 1979. In this algorithm, called (t, n)-threshold scheme [9], [12], [18], the author assumes that a trusted entity, called the *dealer*, uses the Horner algorithm to divide a secret s into n shares $s_i = (x_i, p(x_i))$, $i = 0, 1, \ldots, n-1$ and distributes them between the participants of the threshold process, where nonzero knots $(x_i)_0^{n-1}$ are pairwise distinct and

$$p(x) = \sum_{i=0}^{t-1} a_i x^i, a_0 = s, t \leq n, \tag{1}$$

is an univariate polynomial of degree $t-1$ over a field $\mathbb{K}$ of order $|\mathbb{K}|$ greater than n. Then owners of any t shares $s_{i_0}, s_{i_1}, \ldots, s_{i_{t-1}}$ can ask another trusted entity, called the *combiner*, to recover the secret s by using the well known Lagrange, Newton or Neville interpolating formulae [15], [16]. If the combiner's attempt succeeds, then the owners gain the desired access to some goods.

The authors [16] and Tassa [20] proposed to use the Hermite interpolation in order to extend Shamir's (t,n)-threshold scheme to a hierarchical (t,n)-threshold secret sharing scheme, which admits priorities of the shares during the reconstruction of the secret s. The hierarchy is achieved by admitting, among all $n \geq t$ shares, the shares with consecutive derivatives of $p \in P_{t-1}$ at confluent knots, which are admissible in Hermite interpolation problems. We note that the sequence $(x_i)_0^{n-1}$ is said to be *admissible* if each its longest subsequence of equal knots consists only consecutive knots, say $x_i, x_{i+1}, \ldots, x_{i+k}$ for some i, k. It is equivalent to the fact that

$$x_i = x_{i-1} = \cdots = x_{i-k_i} \text{ for all } i, \tag{2}$$

where k_i are the left multiplicities of x_i defined by

$$k_i = \max\{k : x_{i-k} = x_i\}. \tag{3}$$

In the hierarchical threshold scheme the dealer uses the generalized Horner algorithm to evaluate n shares of the Hermite type,

$$s_i = (k_i, x_i, y_i), \quad y_i = p^{(k_i)}(x_i), \quad i = 0, 1, \ldots, n-1, \tag{4}$$

which means that the sequence $(x_i)_0^{n-1}$ is admissible. Next, he distributes these shares among participants of the threshold scheme. On the other hand, if the combiner receives t shares $s_{i_0}, s_{i_1}, \ldots, s_{i_{t-1}}$ from a coalition of the participants, then he sorts them to get a sequence of shares of the Hermite type, for the simplicity of notation say $(s_i)_0^{t-1}$. After successful sorting the combiner can apply the generalized Lagrange, Newton or Neville forms of the Hermite interpolating polynomial $\tilde{p} \in \mathscr{P}_{t-1}$ such that

$$\tilde{p}^{(k_i)}(x_i) = y_i, \quad i = 0, 1, \ldots, t-1, \tag{5}$$

in order to compute the secret $\tilde{s} = \tilde{p}(0)$, which requires $O(t^2)$ algebraic operations.

Example 1. If $n = 7$, $t = 2$ and

$$(s_i)_0^6 = ((0,5,89), (1,5,65), (0,4,72), (0,8,87), (1,8,69), (2,8,13)),$$

then the combiner's shares $(s_{i_j})_0^2 = ((1, 8, 69), (0, 5, 89), (0, 8, 87))$ can be sorted to the following shares $(s_i)_0^2 = ((0, 8, 87), (1, 8, 69), (0, 5, 89))$ of the Hermite type. On the other hand, such a sorting is impossible for shares $(s_{i_j})_0^1 = ((0, 4, 72), (1, 5, 65))$.

Note that we have $p = \tilde{p}$, and so $s = \tilde{s}$, whenever the shares $(s_i)_0^{t-1}$ are not falsified. Therefore, one can assume that the combiner should grant the access only if $s = \tilde{s}$. Moreover, note that the Shamir's (t, n)-threshold scheme can be adapted to design useful broadcast protocols [3], [5], [8], [10], [11], which include not only protocols for the paid satellite and internet transmissions and transfers for money from banks, but also payments from the automated telling machines. This idea is due to Naor and Pincas [10], [11], who noticed that the group controller (*gc*-unit) of broadcasting plays the similar role as a dealer and a combiner during a secret sharing. Recall that the *gc*-unit is responsible [4] for the following activities:

(i) computation, distribution and reconstitution of the private keys (shares) among the potential recipients of broadcasted messages,

(ii) regular exchange of the session key, in order to identify and revoke illegal users and to add new clients,

(iii) encryption of the session and private keys to increase the security of protocols,

(iv) transmission of the information or some other goods to authorized clients.

In view of (iii) we choose a multiplicative group $\mathbb{G}$ such that, for each $r \in \mathbb{K}\setminus\{0\}$, there exists a function $e_r : \mathbb{K} \to \mathbb{G}$, which satisfies the following functional equations

$$e_r(x) = x, \quad e_r(x + y) = e_r(x)e_r(y), \quad e_r(xy) = [e_r(x)]^y \tag{6}$$

for all $x, y \in \mathbb{K}$. Thus e_r is a homorphism of the additive group $\mathbb{K}$ onto the multiplicative group $\mathbb{G}$. For example, we may use the characters e_r to encrypt both shares and keys, whenever either $\mathbb{K} = \mathbb{Z}_n$ is the additive group of residues modulo a prime number n and $\mathbb{G} = (\mathbb{Z}_n\setminus\{0\}, \cdot)$, or $\mathbb{K} = \mathbb{R}$ is the additive group of real numbers and $\mathbb{G}$ is the multiplicative group of complex numbers of absolute value 1. In these particular cases we can define the characters by

$$e_r(x) = \omega^{rx} \text{ and } e_r(x) = e^{irx},$$

where ω is a generator of the multiplicative group $\mathbb{Z}_n\setminus\{0\}$.

In this paper the encrypted shares of the form

$$s_i = s_i(r) = (k_i, x_i, e_r(y_i)), \quad r \in \mathbb{K}\setminus\{0\}, \quad y_i = p^{(k_i)}(x_i), \tag{7}$$

are used to define the encrypted hierarchical (t, n)-threshold schemes in Section 2. In Section 3 such shares s_i with $k_i = 0$, called the encrypted users' private keys, play a fundamental role in the definition of secure broadcasting protocols. A presentation of these definitions needs to introduce additional details about Hermite interpolation. Namely, throughout the paper the identity of the form

$$(d_i)_0^{t-1} = (d_{\alpha,\kappa})_{\alpha=0\kappa=0}^{m-1\tau_\alpha-1}, \ t = \sum_{\alpha=0}^{m-1} \tau_\alpha, \tag{8}$$

will mean that the lexicographic ordering of the matrix elements on the right hand side is identical with the sequence coordinates on the left hand side.

By using this convention one can divide the sequence $(s_i)_0^{t-1}$ of Hermite type into the sequence of blocks $(S_\alpha)_0^{m-1}$ such that the knots are the same in each block and distinct in any two different blocks:

$$S_\alpha = \{(z_\alpha, \tau_\alpha, y_{\alpha,\kappa}) : \ \kappa = 0, 1, \ldots, \tau_\alpha - 1\},$$

where z_α denotes the common knot in the group S_α and the multiplicities τ_α are defined by

$$\tau_\alpha = card(S_\alpha) = \max\{k_i + 1 : k_i \in S_\alpha\}. \tag{9}$$

Then the Hermite interpolating formula

$$\tilde{p}(x) = \sum_{\alpha=0}^{m-1} \sum_{\kappa=0}^{\tau_\alpha-1} y_{\alpha,\kappa} g_{\alpha,\kappa}(x), \tag{10}$$

with Hermite fundamental polynomials

$$g_{\alpha,\kappa}(x) = \frac{w(x)}{\kappa!} \sum_{\nu=0}^{\tau_\alpha-\kappa-1} \frac{\gamma_{\alpha,\nu}}{(x - z_\alpha)^{\tau_\alpha-\kappa-\nu}}, \tag{11}$$

can be easily proved for the polynomial $\tilde{p} \in \mathscr{P}_{t-1}$, defined by interpolating conditions

$$\tilde{p}^{(\kappa)}(z_\alpha) = y_{\alpha,\kappa}, \ \alpha = 0, 1, \ldots, m-1, \ \kappa = 0, 1, \ldots, \tau_\alpha - 1,$$

which are equivalent to the conditions (5), whenever we denote

$$(y_i)_0^{t-1} = (y_{\alpha,\kappa})_{\alpha=0\kappa=0}^{m-1\tau_\alpha-1}.$$

The representation (10)-(11) of $\tilde{p}(x)$ is called the barycentric Hermite formula, in which the barycentric weights are defined by

$$\gamma_{\alpha,\rho} = \frac{h_\alpha^{(\tau_\alpha-1-\rho)}(z_\alpha)}{(\tau_\alpha-1-\rho)!}, \frac{1}{h_\alpha(x)} = w_\alpha(x) = \prod_{\nu=0,\nu\neq\alpha}^{m-1} (x-z_\nu)^{\tau_\nu} \tag{12}$$

for $\alpha = 0, 1, \ldots, m-1$ and $\rho = 0, 1, \ldots, \tau_\alpha - 1$. It follows that $\gamma_{\alpha,\rho}$ are the unique coefficients of the following partial fraction decomposition

$$\frac{1}{w(x)} = \sum_{\alpha=0}^{m-1} \sum_{\rho=0}^{\tau_\alpha-1} \frac{\gamma_{\alpha,\rho}}{(x-z_\alpha)^{\rho+1}}, \; w(x) = \prod_{\alpha=0}^{m-1} (x-z_\alpha)^{\tau_\alpha}, \tag{13}$$

which is a particular case of (10) for $\tilde{p}(x) \equiv 1$ and $y_{\alpha,\kappa} \equiv \tilde{p}^{(\kappa)}(z_\alpha)$. Further, it should be noticed that the barycentric weights $(\gamma_{\alpha,\rho})_{\alpha=0\rho=0}^{m-1\tau_\alpha-1}$ depend only on the knots $z_\alpha \in S_\alpha$ and its multiplicities $\tau_\alpha = card(S_\alpha)$.

At the present time a few efficient $O(t^2)$-algorithms for computing barycentric weights $\gamma_{\alpha,\rho}$ are known [14], [2], [13]. It should be mentioned that the algorithm of Schneider and Werner [14] transforms numerically the Hermite interpolating polynomial given in the Newtonian form to its barycentric Hermite form. On the other hand, the Butcher's barycentric algorithm is based on some properties of symmetric forms. Clearly, all these algorithms may be used in our Algorithms 2 for secret sharing and broadcasting presented in Section 2 and 3. However, if the knots are defined by the recurrent formulae of the form

$$x_i = \xi x_{i-1} + \beta, \quad i = 1, 2, \ldots, t-1; x_0 = \eta,$$

with $\xi \neq 0, \beta, \eta$ in $\mathbb{K}$, then we propose to use the fast $O(t \log t)$-algorithms presented by the authors in [17]. Finally, in Section 3 a special barycentric algorithm is designed by the second author in order to increase security and effectiveness and generalize broadcasting protocol from the thesis [6].

3. Encrypted hierarchical $(t, n)-$threshold schemes.

Following [16] our first mathematical model of encrypted hierarchical $(t, n)-$threshold scheme is based on the Newton formulae

$$\tilde{p}(x) = \sum_{i=0}^{t-1} c_i h_i(x), \quad c_i = \langle y_0, y_1, \ldots, y_i \rangle, \tag{14}$$

for the Hermite interpolating problem (5), where the Newton basis $(h_i)_0^{t-1}$ of $\mathscr{P}_{t-1}$ is defined by

$$\begin{aligned} &h_0(x) = 1, \\ &h_i(x) = h_{i-1}(x) \cdot (x - x_{i-1}) \\ &\qquad = (x - x_0) \ldots (x - x_{i-1}), \quad i = 1, \ldots, t-1. \end{aligned} \tag{15}$$

It is important in applications that the divided differences satisfy the following useful recurrent formulae

$$\langle y_i, y_{i+1}, \ldots, y_{i+k} \rangle = \begin{cases} \frac{y_{i+k-k_i}}{(i+k-k_i)!}, & \text{if } x_i = x_{i+k}, \\ \frac{\langle y_{i+1}, \ldots, y_{i+k} \rangle - \langle y_i, \ldots, y_{i+k-1} \rangle}{x_{i+k} - x_i}, & \text{otherwise}, \end{cases} \tag{16}$$

for all i, k such that $0 \le i \le i + k < t$. Hence the coefficients $(c_i)_0^{t-1}$ can be computed by using the following slight modification of the effective $O(t^2)-$algorithm [19], which should be proceeded by the initializations given at the first line of the algorithm.

```
b_i = y_i/k_i!,   i = 0, 1, ..., t − 1;
for(i = 0; i < t; i++)
{
    for(j = i; j >= i − k_i; j−−)
        b_j = b_i;
    for(j = i − k_i − 1; j >= 0; j−−)
        b_j = (b_{j+1} − b_j)/(x_i − x_j);
    c_i = b_0;
}
```

Algorithm DD. The C++ pseudocode to compute vectors c and b of divided differences $c_i = \langle y_0, y_1, \ldots, y_i \rangle$ and $b_i = \langle y_i, y_{i+1}, \ldots, y_{t-1} \rangle$, $i = 0, 1, \ldots, t-1$.

Now by using formulae (14) and (15) we derive the formula

$$\tilde{s} = \tilde{p}\,(0) = \sum_{i=0}^{t-1} c_i h_i(0) \tag{17}$$

with

$$h_0(0) = 1,\ h_i(0) = -h_{i-1}(0)x_{i-1},\ i = 1, 2, \ldots, t-1.$$

It suggests the following simple hierarchical (t, n)-threshold scheme with an arbitrary encryption function e_r, which do not need to have properties (6). However, it is supposed that this function is known only by the dealer, while the decryption function e_r^{-1} is known only by the combiner. Note that if we use the encryption function $e_r(x) = \omega^{rx}$ $(x \in \mathbb{Z}_n)$ in Algorithm 1, then the security of the $(t, n)-$threshold scheme increases proportionally to the high security of the Diffi-Helman decision problem [1]. The overall number of algebraic operations in Algorithm 1 is equal to $O(t^2)$, whenever evaluations of the encryption and decryption functions are not taken into account.

Algorithm 1. Dividing the secret s into n encrypted hierarchical (t, n)-shares and its recovering. The encryption and decryption functions e_r and e_r^{-1}, $r \in \mathbb{K}\setminus\{0\}$, are supposed to be known by the dealer and combiner, respectively.

1. The dealer chooses randomly two integers $n \geq t > 1$, admissible Hermite interpolation knots $(x_i)_0^{n-1}$ in $\mathbb{K}\setminus\{0\}$ with left multiplicities $(k_i)_0^{n-1}$, coefficients $(a_i)_1^{t-1}$ of the polynomial $p(x) = \sum_{i=0}^{t-1} a_i x^i$ of degree $t-1$, and sets $a_0 = s$.

2. Next the dealer uses the generalized Horner algorithm to compute n values/derivatives $y_i = p^{(k_i)}(x_i)$ of the polynomial $p(x)$, encrypts them by setting $u_i = e_r(y_i)$ and distributes the encrypted shares $s_i(r) = (k_i, x_i, u_i)$, $i = 0, 1, \ldots, n-1$, among the participants of the scheme.

3. If the combiner receive t shares $s_{i_0}(r)$, $s_{i_1}(r)$, $\ldots$, $s_{i_{t-1}}(r)$, then he sorts them in order to get a sequences of the Hermite type, say $(k_i,\ x_i,\ u_i)_0^{t-1}$ for the simplicity of notation. If such a sorting is impossible, the algorithm is stopped without granting the access.

4. Otherwise, the combiner applies the decryption function to get $y_i = e_r^{-1}(u_i)$ and uses Algorithm DD to compute the divided differences $c_i =< y_0, \ldots, y_i >$,

$i = 0, 1, \ldots, t-1$. It enables to compute the secret $\tilde{s} = \tilde{p}(0)$ from the formula (17), or equivalently from the recurrent formulae: $\tilde{s} = 0$ and $\tilde{s} = c_i - \tilde{s} \cdot x_i$, $i = t-1, t-2, \ldots, 0$.

5. The access is granted only if $\tilde{s} = s$.

Our second mathematical model of an encrypted $(t, n)-$threshold scheme depend heavily on the properties (6) of the encryption function e_r, $r \in \mathbb{K}\backslash\{0\}$. It is based on the following theorem, in which the convention (8) is applied to exchange lexicographically the double indexed variables $y_{\alpha,\kappa}$, $g_{\alpha,\kappa}$, $\gamma_{\alpha,\rho}$ $(0 \leq \alpha < m, 0 \leq \kappa, \rho < \tau_\alpha)$ into the corresponding single indexed variables y_i, g_i, γ_i $(0 \leq i < t)$. Moreover, it is supposed that the definition (9) of the multiplicity τ_α of z_α is extended by setting $\tau_i = \tau_\alpha$, whenever $x_i = z_\alpha$. For example, if we consider the admissible knots $(x_i)_0^6 = (5, 5, 4, 8, 8, 8)$ with the left multiplicities $(k_i)_0^6 = (0, 1, 0, 0, 1, 2)$, then the last extension gives $(\tau_i)_0^6 = (2, 2, 1, 3, 3, 3)$.

Theorem 1 *Let $s_i(r) = (k_i, x_i, e_r(y_i))$ $(i = 0, 1, \ldots, t-1)$ be shares of the Hermite type, derived from the polynomial p of the form (1) with $a_0 = e_r^{-1}(s)$, and encrypted by the function e_r having properties (6). Moreover, let the polynomial $\tilde{p} \in \mathscr{P}_{t-1}$ be defined by Hermite interpolating conditions $\tilde{p}^{(k_i)}(x_i) = y_i$ $(i = 0, 1, \ldots, t-1)$ and expressed in the Hermite barycentric form with barycentric weights $(\gamma_i)_0^{t-1}$. Then $\tilde{s} = e_r(\tilde{p}(0))$ satisfies the following formula*

$$\tilde{s} = \prod_{i=0}^{t-1} e_r(y_i)^{g_i(0)}, \tag{18}$$

with

$$g_i(0) = \frac{w(0)}{k_i!} \sum_{\nu=0}^{\tau_i - k_i - 1} \frac{\gamma_\nu}{(-x_i)^{\tau_i - k_i - \nu}}, \quad w(0) = \prod_{i=0}^{t-1} (-x_i), \tag{19}$$

where $(k_i)_0^{t-1}$ and $(\tau_i)_0^{t-1}$ denote the left multiplicities and multiplicities of knots $(x_i)_0^{t-1}$, respectively. Further, the cost of computation of $(g_i(0))_0^{t-1}$ is equal to $O(t^2)$.

Proof. In view of properties (6) of encryption function e_r, it follows from formulae (10) and (11) that

$$e_r(\tilde{p}(0)) = \prod_{\alpha=0}^{m-1} \prod_{\kappa=0}^{\tau_\alpha - 1} (e_r(y_{\alpha,\kappa}))^{g_{\alpha,\kappa}(0)}$$

and

$$g_{\alpha,\kappa}(0) = \frac{w(0)}{\kappa!} \sum_{\nu=0}^{\tau_\alpha-\kappa-1} \frac{\gamma_{\alpha,\nu}}{(-z_\alpha)^{\tau_\alpha-\kappa-\nu}}.$$

Hence it is sufficient to pass lexicographically from the double to single indexes in order to get formulae (18) and (19). Since the barycentric weights can be evaluated by an $O(t^2)$-algorithm, it remains to show that the cost of computing of all basis sums

$$\sigma_{\alpha,\kappa} = \sum_{\nu=0}^{\tau_\alpha-\kappa-1} \frac{\gamma_{\alpha,\nu}}{(-z_\alpha)^{\tau_\alpha-\kappa-\nu}}, \quad \kappa = 0, 1, \ldots, \tau_\alpha - 1, \tag{20}$$

is equal to $O(\tau_\alpha)$. But it is a simple consequence of Algorithm BS, which is obtained below by applying the modified Horner algorithm to compute the basis sums in the order $\sigma_{\alpha,\tau_\alpha-1}, \sigma_{\alpha,\tau_\alpha-2}, \ldots, \sigma_{\alpha,0}$. □

In the prove of Theorem 1 the basis sums (20) have been defined. It is essential that they can be computed by the Algorithm BS, which has only the cost $O(\tau_\alpha)$. Note that this algorithm computes the basis sums in order $\sigma_{\alpha,\tau_\alpha-1}, \sigma_{\alpha,\tau_\alpha-2}, \ldots, \sigma_{\alpha,0}$ by the Horner algorithm with divisions.

$\sigma_{\alpha,\tau_\alpha} = 0;$
$for(\nu = \tau_\alpha - 1; \nu \geq 0; \nu - -)$
$\quad \sigma_{\alpha,\nu} = (\sigma_{\alpha,\nu+1} + \gamma_{\alpha,\tau_\alpha-\nu-1})/(-z_\alpha);$

Algorithm BS. Evaluation of the basis sums (20) with $O(\tau_\alpha)$ algebraic operations.

Now we are ready to design a mathematical model of a new hierarchical (t, n)-threshold scheme with an encryption e_r having properties (6). In order to compute the barycentric weights $(\gamma_i)_0^{t-1} = (\gamma_{\alpha,\rho})_{\alpha=0,\rho=0}^{m-1,\tau_\alpha-1}$ we propose to use one of the algorithms presented in [14], [2], [13].

Algorithm 2. A barycentric recovering of the secret s divided into n encrypted hierarchical (t, n)-shares. The encryption and decryption functions e_r and e_r^{-1}, $r \in \mathbb{K}\setminus\{0\}$, are supposed to be known only by the dealer.

1. Here the dealer's duties are the same as in the Step 1 of Algorithm 1, except the fact that he should set $a_0 = e_r^{-1}(s)$ now.

2. Next the dealer generates and distributes the shares $s_i(r) = (k_i, x_i, \ u_i)_0^{n-1}$ with $u_i = e_r(y_i)$, exactly as in the Step 2 of Algorithm 1.

3. If the combiner receive t shares $s_{i_0}(r)$, $s_{i_1}(r)$, ..., $s_{i_{t-1}}(r)$, then he sorts them in order to get a sequences of the Hermite type, say $(k_i, x_i, \ u_i)_0^{t-1}$ for the simplicity of notation. If such a sorting is impossible, the algorithm is stopped without granting the access.

4. Otherwise, the combiner uses left multiplicities $(k_i)_0^{t-1}$, multiplicities $(\tau_i)_0^{t-1}$ and knots $(x_i)_0^{t-1}$ in order to compute the barycentric weights $(\gamma_i)_0^{t-1}$ and evaluate the basis values $(g_i(0))_0^{t-1}$ with the help of Algorithm BS. Then he computes $\tilde{s}$ from the formulae (18) presented in Theorem 1.

5. The access is granted only if $\tilde{s} = s$.

4. Barycentric broadcast protocol with encryption

In this section we present a general broadcast model that uses the barycentric form of the Hermite interpolating polynomials and the encryption function e_r with properties (6). It is based on the following theorem, which will reflect the dynamic of broadcast cycles.

Theorem 2 *Let $(\gamma_{\alpha,\rho})_{\alpha=0}^{m-1}{}_{\rho=0}^{\tau_\alpha-1}$ be barycentric weights, determined by a sequence $(z_\alpha)_0^{m-1}$ of pairwise distinct knots with multiplicities $(\tau_\alpha)_0^{m-1}$. If $z_m \notin \{z_0, z_1, \ldots, z_{m-1}\}$ is an additional knot of multiplicity $\tau_m = 1$, then the barycentric weights $(\hat{\gamma}_{\alpha,\rho})_{\alpha=0}^{m}{}_{\rho=0}^{\tau_\alpha-1}$, corresponding to sequences $(z_\alpha)_0^m$ and $(\tau_\alpha)_0^m$, satisfy the following recurrent formulae*

$$\hat{\gamma}_{\alpha,\tau_\alpha-1} = \frac{\gamma_{\alpha,\tau_\alpha-1}}{z_\alpha - z_m},$$
$$\hat{\gamma}_{\alpha,\tau_\alpha-\kappa} = \frac{\gamma_{\alpha,\tau_\alpha-\kappa} - \hat{\gamma}_{\alpha,\tau_\alpha-\kappa+1}}{z_\alpha - z_m}, \quad \kappa = 2, 3, \ldots, \tau_\alpha,$$

where $\alpha = 0, 1, \ldots, m-1$. Additionally, we have $\hat{\gamma}_{m,0} = \frac{1}{w(z_m)}$, where the polynomial $w(x)$ is defined as in (13).

Proof. It is clear, that $\hat{\gamma}_{m,0} = \hat{h}_m(z_\alpha) = \frac{1}{w(z_m)}$. Moreover by (12) we have

$$\hat{\gamma}_{\alpha,\tau_\alpha-\kappa} = \frac{\hat{h}_\alpha^{(\kappa-1)}(z_\alpha)}{(\kappa-1)!}, \quad \hat{h}_\alpha(x) = h_\alpha(x) \cdot \frac{1}{x - z_m},$$

whenever $0 \le \alpha < m$. Hence we obtain the formula for $\hat{\gamma}_{\alpha,\tau_\alpha-1}$ by setting $\kappa = 1$. Otherwise, we apply Leibnitz differentiation rule and formulae (12) to get

$$\hat{\gamma}_{\alpha,\tau_\alpha-\kappa} = \frac{1}{(\kappa-1)!}\sum_{i=0}^{\kappa-1}\binom{\kappa-1}{i}h_\alpha^{(\kappa-1-i)}(z_\alpha)\frac{(-1)^i i!}{(z_\alpha - z_m)^{i+1}}$$
$$= \sum_{i=0}^{\kappa-1}(-1)^i\frac{\gamma_{\alpha,\tau_{\alpha-\kappa+i}}}{(z_\alpha - z_m)^{i+1}}.$$

By using twice this formula we derive

$$\hat{\gamma}_{\alpha,\tau_\alpha-\kappa} + \frac{\hat{\gamma}_{\alpha,\tau_\alpha-(\kappa-1)}}{z_\alpha - z_m} = \frac{\gamma_{\alpha,\tau_\alpha-\kappa}}{z_\alpha - z_m} + \sum_{i=1}^{\kappa-1}(-1)^i\frac{\gamma_{\alpha,\tau_\alpha-\kappa+i}}{(z_\alpha - z_m)^{i+1}}$$
$$+ \frac{1}{z_\alpha - z_m}\sum_{i=0}^{\kappa-2}(-1)^i\frac{\gamma_{\alpha,\tau_\alpha-\kappa+i+1}}{(z_\alpha - z_m)^{i+1}} = \frac{\gamma_{\alpha,\tau_\alpha-\kappa}}{z_\alpha - z_m},$$

which completes the proof. □

A simple consequence of Theorem 2 is the following Algorithm BW, which has the cost $O(t)$. It is an essential part of our new broadcasting protocol with encryption.

$\hat{\gamma}_{\alpha,\tau_\alpha} = 0;$
$for(\kappa = 1; \kappa \le \tau_\alpha; \kappa++)$
$\quad \hat{\gamma}_{\alpha,\tau_\alpha-\kappa} = \frac{\gamma_{\alpha,\tau_\alpha-\kappa} - \hat{\gamma}_{\alpha,\tau_\alpha-\kappa+1}}{z_\alpha - z_m};$

Algorithm BW. Evaluation of the barycentric weights $(\hat{\gamma}_{\alpha,\kappa})_{\kappa=0}^{\tau_\alpha}$, whenever a new knot z_m of multiplicity $\tau_m = 1$ is added to the knots $(z_\alpha)_0^{m-1}$ with multiplicities $(\tau_\alpha)_0^{m-1}$.

Now we present a (t,n)-generalization of the broadcast model of Naor and Pincas [10]. It uses the barycentric weights $(\hat{\gamma}_i)_0^t = (\hat{\gamma}_{\alpha,\kappa})_{\alpha=0\ \kappa=0}^{m\ \ \tau_\alpha-1}$, presented in Theorem 2, to reconstruct the secret s from the formulae

$$\hat{s} = \prod_{i=0}^{t}(e_r(y_i))^{\hat{g}_i(0)} \tag{21}$$

and

$$\hat{g}_i(0) = \frac{\hat{w}(0)}{k_i!}\sum_{\nu=0}^{\tau_i-k_i-1}\frac{\hat{\gamma}_\nu}{(-x_i)^{\tau_i-k_i-\nu}}, \quad \hat{w}(0) = \prod_{i=0}^{t}(-x_i), \tag{22}$$

where $\tau_m = 1$, $x_t = z_m$ and $k_t = 0$. These formulae can be obtained formally from Theorem 1 by exchanging $t-1$ by t and by adding value $y_t = p(x_t)$ of p at the point $x_t \notin \{x_0, x_1, \ldots, x_{t-1}\}$. The description of this generalization is divided into the system initialization and the broadcast cycle.

Algorithm 3. A barycentric (t, n)-broadcast model with a secret key $s \in \mathbb{K}\setminus\{0\}$ and encryption function e_r, $r \in \mathbb{K}\setminus\{0\}$.

1. In order to initialize the model the group controller randomly choses the coefficients $(a_i)_1^t$ of the polynomial $p(x) = \sum_{i=0}^{t} a_i x^i$, $a_0 = e_r^{-1}(s)$, of degree t, and admissible knots $(x_i)_0^{n-1}$ with the left multiplicities $(k_i)_0^{n-1}$ such that $k_i = 0$ for $i = t, t+1, \ldots, n-1$. Then he initializes the list $L = \emptyset$ of revoked users, and passes to:
   (a) Compute the derivatives/values $y_i = p^{(k_i)}(y_i)$ at points x_i ($i = 0, 1, \ldots, n-1$) and deliver $n-t$ encrypted shares $(x_i, e_r(y_i))_t^{n-1}$ to the users, which are called users keys.
   (b) Evaluate the public session key $\Gamma = (\gamma_i)_0^{t-1}$ of barycentric weights by applying any known $O(t^2)$-algorithm to the knots $(x_i)_0^{t-1}$ with left multiplicities $(k_i)_0^{t-1}$.
   (c) Prepare the public encrypted enabling block E of the following form $E = (k_i, x_i, e_r(y_i))_0^{t-1}$.
2. A broadcast cycle begins at the moment, when a private user's key $(x_s, e_r(y_s))$, $s \geq t$, has been received by the system, together with a (promise of) payment for a required transmission. Then the following activities are performed:
   (a) The user's key $(x_s, e_r(y_s))$ is reindexed and used to update the enabling block E to $\hat{E} = (k_i, x_i, e_r(y_i))_0^t$ ($k_t = 0,\ x_t := x_s$).
   (b) The Algorithm BW is applied to update the barycentric weights $\Gamma = (\gamma_i)_0^{t-1}$ to $\hat{\Gamma} = (\hat{\gamma}_i)_0^t$, which correspond to the knots $(x_i)_0^t$ with left multiplicities $(k_i)_0^t$ and multiplicities $(\tau_i)_0^t$ defined as in (9).
   (c) The formulae (21) and (22) are used to compute a candidate $\hat{s}$ for s.
   (d) The access to the transmission takes place only if $\hat{s} = s$. Otherwise, the private user's key may be added to the list of revoked users, which may be checked in the next cycles of the algorithm e.g. at the beginning of Stage II.

Since the session key Γ and enabling block E are public, it follows that the second part of Algorithm 3 can be performed not only by the group controller but by any user as well. In the last case, it is necessary to assume that each user is able to activate the algorithms to evaluate the barycentric weights $(\hat{\gamma}_i)_0^t$ and decoding key $\hat{s}$. Of course, it does not mean that knowledge of $\hat{s}$ by a user is allowed in Algorithm 3.

References

[1] D. Boneh (1998), The decision Diffie - Hellman problem, Lecture Notes in Computer Science 1423, 48 - 63.

[2] J.C. Butcher, R.M. Corless, L. Gonzalez-Vega, A. Shakoori (2011), Polynomial algebra for Birkhoff interpolants. Numerical Algorithms 56, 319 - 347.

[3] V. Daza, J. Herranz, P. Morillo, C. Ráfols (2008), Ad-hoc threshold broadcast encryption with shorter ciphertexts, Electronic Notes in Theoretical Computer Science 192 (2), 3 - 15.

[4] P. Feldman (1987), A practical scheme for non-interactive verifiable secret sharing, IEEE Symposium on Foundations of Computer Science, 427 - 437.

[5] A. Fiat, M. Naor (1993), Broadcast encryption, Lecture Notes in Computer Science 773, 480 - 491.

[6] J. Kapusta (2010), Algorithms for polynomial transformation and its applications, Ph. D. dissertation, System Research Institute Polish Academy of Sciences, in Polish.

[7] N. Kogan, T. Tassa (2006), Improved efficiency for revocation schemes via Newton interpolation, ACM Transactions on Information and System Security 9, 461 - 486.

[8] L. Krzywiecki, M. Kutylowski, M. Nikodem (2007), General anonymous key broadcasting via Lagrangian interpolation , 1st International Workshop on Group-Oriented Cryptographic Protocols, IET Information Security 2 (3), 79 - 84.

[9] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone (2005), Handbook of Applied Cryptography.

[10] M. Naor, B. Pinkas (2000), Efficient trace and revoke schemes, Financial Cryptography 2000. Lecture Notes in Computer Science 1962, 1 - 20.

[11] M. Naor, B. Pinkas (2010), Efficient trace and revoke schemes, International Journal of Information Security 9 (6), 411 - 424.

[12] J. Pieprzyk, T. Hardjono, J. Seberry (2003), Fundamentals of Computer Security, Springer Verlag.

[13] B. Sadiq, D. Viswanath (2013), Barycentric Hermite Interpolation, Numerical Analysis, SIAM Journal on Scientific Computing 35, 1254 - 1270.

[14] C. Schneider, W. Werner (1991), Hermite interpolation: The barycentric approach, Computing 46 (1), 35 - 51.

[15] A. Shamir (1979), How to share a secret, Communications of the ACM 22 (11), 612 - 613.

[16] R. Smarzewski, J. Kapusta (2005), Algorithms for multi-secret hierarchical sharing schemes of Shamir type, Annales UMCS Informatica AI III, 65 - 91.

[17] R. Smarzewski, J. Kapusta (2007), Fast Lagrange-Newton transformations, Journal of Complexity 23 (3), 336 - 345.

[18] D. R. Stinson (1995), Cryptography Theory and Practice, CRC Press.

[19] J. Stoer, R. Bulirsch (2002), Introduction to Numerical Analysis, Springer.

[20] T. Tassa (2007), Hierarchical Threshold Secret Sharing, Journal of Cryptology 20 (2), 237 - 264.

[21] W. Tzeng, H. Tzeng (2005), A public-key traitor tracing scheme with revocation using dynamic shares, Designs, Codes and Cryptography 35, 47 - 61.