Identity-Based Threshold Signature Scheme from the Bilinear Pairings (Extended Abstract)

Joonsang Baek
Monash University, Australia
E-mail:joonsang@infotech.monash.edu.au

Yuliang Zheng
UNC Charlotte, USA
E-mail:yzheng@uncc.edu

Abstract

The focus of this paper is to formalize the concept of identity-based threshold signature and give the first provably secure scheme based on the bilinear pairings. An important feature of our scheme is that a private key associated with an identity rather than a master key of the Public Key Generator is shared among signature generation servers, which is more desirable in practice. Another interesting aspect of our results is that the security of one of the verifiable secret sharing schemes used to construct the identity-based threshold signature scheme is relative to a slightly modified version of the Generalized Tate Inversion problem recently proposed by Joux.

Keywords: Identity-based threshold signature, Bilinear Pairing, Verifiable secret-sharing, Generalized Bilinear Problem, Chosen message attack

1. Introduction

Motivation. Threshold signature is a useful tool for decentralizing the power to sign a message. A major motivation for identity (ID)-based signature originally proposed by Shamir [15] is to authenticate messages without the need of exchanging public keys or keeping public key directories. A major advantage of ID-based signature is that it allows one to sign a message in such a way that any user can verify the signature using the signer's identifier such as email address instead of using his/her digital certificate. Combining these two concepts to realize "ID-based threshold signature" is the focus of this paper.

A practical application of ID-based threshold signature scheme can be considered in the following situation. Suppose that Alice, as a president of a company, has created an identity which represents the company and has a private key associated with the identity. Using the private key, she is able to sign any documents. But (concerning about, e.g, the situation where she is away) she wants to delegate this power to a number of signature-generation servers so that a signature on a given message is jointly generated by those servers and anyone can successfully verify the signature using the company's published identity if, and only if, he/she obtains a certain number of partial signatures from the signature-generation servers.

Due to the page limit, discussions on related work, security notions and analyses of the various verifiable secret-sharing and ID-based threshold signature schemes are brief. Readers are referred to [2] for full details.

Related Work. In the non-ID-based setting, research on threshold signature schemes has been quite active. The notable contributions in this line of research especially focused on the formalizations of threshold signature and its security include the works of Cerecedo, Matsumoto and Imai [5] and those of Gennaro, Jarecki, Krawczyk, and Rabin's [9].

Since Boneh and Franklin [4] used the bilinear pairings to construct the first functional ID-based encryption scheme, several ID-based signature schemes based on the bilinear pairings including Cha and Cheon's scheme [6] and Hess' scheme [11] have been proposed. The authors of [6] and [11] gave a formal definition of the unforgeability of ID-based signature schemes against chosen message attack, and proved their schemes are secure in the random oracle model [3] assuming the Computational Diffie-Hellman (CDH) problem is intractable.

However, to our knowledge, ID-based threshold signature has not been treated in the literature.

Our Contribution. In this paper, we present a formal security notion for ID-based threshold signature and give a concrete scheme whose unforgeability against chosen message attack is based on the CDH problem. Interestingly, the security of one of the verifiable secret sharing schemes that is used to construct our ID-based threshold scheme is relative to a variant of the

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

Generalized Tate Inversion (GTI) problem. Joux asked the question whether the problem can be used to construct cryptographic protocols [12]. Our results answer the question in an affirmative way.

2. Security Notion of ID-Based Threshold Signature

Generic ID-based Threshold Signature. We first describe a generic (*t, n*) ID-based threshold signature scheme, which we call "$\mathcal{IDTHS}$". $\mathcal{IDTHS}$ consists of algorithms GC, EX, DK, S, and V.

The common parameter generation algorithm GC is run by the trusted Private Key Generator (PKG) to generate its master/public key pair and all the necessary common parameters. The PKG's public key and the common parameters are given to every interested party.

On receiving a user's private key extraction request which consists of an identity, the PKG then runs the private key extraction algorithm EX to generate the private key associated with the requested identity.

An authorized dealer who possesses a private key associated with an identity runs the private key distribution algorithm DK to distribute the private key to *n* signature generation servers. The entity that runs DK can be either a normal user (such as Alice in the example given in Section 1) or the PKG, depending on cryptographic services that the PKG can offer. Namely, if the PKG is able to organize threshold signature, the PKG can run DK, but if the PKG has the only functionality of generating private keys for users, the entity running DK would be a trusted normal user. Note that like the ID-based threshold decryption scheme proposed in [1], the private key associated with an identity is shared in our ID-based threshold signature scheme. According to [1], this approach is more practical than Boneh and Franklin [4]'s approach that distributes the master key of the PKG to a number of other PKGs (called "Distributed PKGs") to perform cryptographic operations such as threshold decryption or signature generation, since the latter approach requires the distributed PKGs to be involved *on-line* in performing the cryptographic operations. Obviously, this creates a bottleneck on the PKGs and also violates one of the basic requirements of an ID-based encryption scheme, "the PKG can be closed after key generation", which was envisioned by Shamir in his original proposal of ID-based cryptography [15].
Moreover, it introduces a scalability problem when the number of available distributed PKGs is not matched against the number of decryption servers required, say, there are only 3 available PKGs while a certain application requires 5 signature generation servers.

Given a set of common parameters generated by GC, a share of a private key associated with an identity, and a message, *n* signature generation servers jointly generate a signature for the given message by running the signature generation algorithm S.

The validity of the signature can be checked by running the signature verification algorithm V.

3. Computational Assumptions

Bilinear Pairing. We briefly review the admissible bilinear pairing [4], which we call the "Bilinear pairing" for short. The Bilinear pairing $\hat{e}$ is defined over two groups of the same prime-order q denoted by $\mathcal{G}$ and $\mathcal{F}$. (By $\mathcal{G}^*$ and Z_q^*, we denote $\mathcal{G}-\{O\}$ where O denotes the identity element of $\mathcal{G}$, and $Z_q^*-\{0\}$ respectively.) We will use an additive notation to describe the operation in $\mathcal{G}$ while we will use a multiplicative notation for the operation in $\mathcal{F}$. In practice, the group $\mathcal{G}$ will be implemented using a group of points on certain supersingular elliptic curves and the group $\mathcal{F}$ will be implemented using a subgroup of the multiplicative group of a finite field. The Bilinear pairing $\hat{e}:\mathcal{G} \times \mathcal{G} \rightarrow \mathcal{F}$ has the following properties [4].

- Bilinear: $\hat{e}(aR_1, bR_2)= \hat{e}(R_1,R_2)^{ab}$, where $R_1, R_2 \in \mathcal{G}$ and $a, b \in Z_q^*$.
- Non-degenerate: $\hat{e}$ does not send all pairs of points in $\mathcal{G} \times \mathcal{G}$ to the identity in $\mathcal{F}$. (Hence, if R is a generator of $\mathcal{G}$ then $\hat{e}(R, R)$ is a generator of $\mathcal{F}$.)
- Computable: For all $R_1, R_2 \in \mathcal{G}$, the pairing $\hat{e}(R_1, R_2)$ is efficiently computable.

CDH Problem in the group $\mathcal{G}$. Informally, the CDH problem in the group $\mathcal{G}$ refers to the following computational problem: Given ($\mathcal{G}$, q, P, aP, bP) for random $a, b \in Z_q^*$, compute $abP \in \mathcal{G}$.

It is believed that the CDH problem in the group $\mathcal{G}$ is computationally intractable. However, it should be noted that the Decisional Diffe-Hellman (DDH) problem in this group can be solved in polynomial time with the help of the Bilinear pairing.

Modified Generalized Bilinear Inversion Problem. Recently, Joux [12] proposed a new computational problem related to the Tate pairing, called the Generalized Tate Inversion (GTI) problem. Informally, the GTI problem refers to the computational problem in which an attacker tries, given random $h \in \mathcal{F}$, to find a pair $(S, T) \in \mathcal{G} \times \mathcal{G}$ such that $e(S, T) = h$, where e denotes the Tate pairing.

We slightly modify the above GTI problem and obtain a new computational problem, which we call a

"modified Generalized Bilinear Inversion (mGBI)" problem". The newly modified computational problem is described as follows: Given random $h \in \mathcal{F}$ and the generator P of $\mathcal{G}$, compute $S \in \mathcal{G}$ such that $\hat{e}(S, P) = h$.

We note that the mGBI assumption (that is, the mGBI problem is computationally intractable) is weaker than the GBI (Generalized Bilinear Inversion) assumption which is naturally derived from the GTI assumption by replacing the Tate pairing by the admissible bilinear pairing, as sketched below: Assume $(\mathcal{G}, \hat{e}, h)$, where $h \in \mathcal{F}$, is given to an attacker $\mathsf{A}^{\mathsf{GBI}}$ for the GBI problem. First, $\mathsf{A}^{\mathsf{GBI}}$ chooses a generator P of $\mathcal{G}$. It then runs an attacker $\mathsf{A}^{\mathsf{mGBI}}$ for the mGBI problem providing $(\mathcal{G}, \hat{e}, h, P)$ as input. When $\mathsf{A}^{\mathsf{mGBI}}$ outputs S, $\mathsf{A}^{\mathsf{GBI}}$ sets $T=P$ and returns (S, T).

4. Building Blocks for Our ID-based Threshold Signature Scheme

Review of "Secret-Sharing over $\mathcal{G}$". In order to share a private key which is associated with an identity, we need the following technique of sharing a point on $\mathcal{G}$ presented in [1].

Distribution Phase: Let q be a prime order of a group $\mathcal{G}$ of points on some elliptic curve. Let $S \in \mathcal{G}^*$ be a secret (point) to share. Suppose that we have chosen integers t and n satisfying $1 \le t \le n < q$.

First, we pick $F_1, F_2, \ldots, F_{t-1}$ uniformly at random from $\mathcal{G}^*$. Then, we define a polynomial-like function F: $\mathbf{N} \cup \{0\} \rightarrow \mathcal{G}$, which we call "PLF" throughout this paper, such that

$$F(x) = S + \sum_{l=1}^{t-1} x^l F_l \cdot$$

Define $t-1$ as a "degree". Now, we compute $S_i = F(i) \in \mathcal{G}$ for $i = 1, \ldots, n$ and send (i, S_i) to the i-th member of the group of cardinality n. Note that when $i=0$, we obtain the secret itself, that is, $S=F(0)$.

Reconstruction Phase: Let $\Psi \subseteq \{1, \ldots, n\}$ be a set such that $|\Psi| \ge t$, where $|\cdot|$ denotes the cardinality of a given set. The function $F(x)$ can be reconstructed by computing

$$F(x) = \sum_{j \in \Psi} \pi_{xj}^{\Psi} S_j \text{ where } \pi_{xj}^{\Psi} = \prod_{\iota \in \Psi, \iota \neq j} \frac{x - \iota}{j - \iota}.$$

Note that $\pi_{xj}^{\Psi} \in Z_q^*$ is the Lagrange interpolation coefficient used in Shamir's secret sharing scheme.

Computationally Secure Verifiable Secret Sharing Scheme Based on the Bilinear Pairing. Verifiable Secret-Sharing (VSS) is a useful tool for providing threshold signature schemes with robustness.

Since our ID-based threshold signature scheme is of Discrete Logarithm (DL)-type, various DL-type VSS schemes, e.g., [7, 13] can be considered as building blocks for our scheme. However, we modify those schemes as the base secret-sharing scheme presented in the previous section, which is different from Shamir's original secret-sharing scheme, and the Bilinear pairing should be employed in our ID-based threshold signature scheme.

Our first VSS scheme, which we call "Computationally secure Verifiable secret-sharing scheme based on the Bilinear Pairing (**CVSSBP**)", is motivated by Feldman's VSS scheme [7]. This scheme will be used to distribute a private key associated with an identity into a number of signature generation servers.

Description of **CVSSBP** Let $(\mathcal{G}, q, P, \hat{e})$ be a set of parameters, as defined previously. Suppose that a threshold t and the number of servers n satisfy $1 \le t \le n < q$. To share a secret $S \in \mathcal{G}^*$ out among n parties, a dealer performs the following steps.

1. Choose $F_1, F_2, \ldots, F_{t-1}$ uniformly at random from $\mathcal{G}^*$, construct a PLF $F(x) = F(x) = S + xF1 + \cdots + x^{t-1}F_{t-1}$ for $x \in \mathbf{N} \cup \{0\}$ and compute $S_i = F(i)$ for $i = 0, \ldots, n$. Note that $S_0 = S$.
2. Send S_i to party Γ_i for $i = 1, \ldots, n$ secretly. Broadcast $\alpha_0 = \hat{e}(S, P)$ and $\alpha_j = \hat{e}(F_j, P)$ for $j = 1, \ldots, t-1$.
3. Each party Γ_i then checks whether its share S_i is valid by computing

$$\hat{e}(S_i, P) = \prod_{j=0}^{t-1} \alpha_j^{i^j} \quad (1)$$

Note that the security of **CVSSBP** is based on the mGBI problem introduced in Section 3. (Readers are referred to [2] for a detailed proof.)

Unconditionally Secure Verifiable Secret Sharing Scheme Based on the Bilinear Pairing. Our second VSS scheme based on the Bilinear pairing, which we call a "Unconditionally-secure VSS based on the Bilinear Pairing (**UVSSBP**)", will be served as a base scheme for the new distributed key generation protocol based on the Bilinear pairing that will be in our ID-based threshold signature scheme. We first define a new commitment scheme.

New Commitment Scheme Let $(\mathcal{G}, q, P, \hat{e})$ be the common parameters, as defined previously. Suppose that random elements $G, H \in \mathcal{G}^*$, and the common parameters are given to a dealer. (We assume that no party knows $a, b \in Z_q^*$ such that $G = aP$ and $H = bP$. These values can be chosen by a trusted third party or interested parties using a coin-flipping protocol.) The dealer then chooses $r \in Z_q^*$ uniformly at random and computes

$$Comm(S, r) = \hat{e}(S, P)\hat{e}(G, H)^r.$$

Description of **UVSSBP** Suppose that the threshold t and the number of parties n satisfy $1 \le t \le n < q$. To share a secret $S \in \mathcal{G}^*$ out among n parties, a dealer performs the following steps.

1. Publish $\delta_0 = Comm(S, r)$, a commitment to S, where r is chosen uniformly at random from Z_q^*. (We assume that the dealer has used the random elements G, H $\in \mathcal{G}^*$ for input parameters for the commitment.)
2. Choose $F_1,\ldots,F_{t-1}$ uniformly at random from $\mathcal{G}^*$, construct a PLF $F(x)=S+ xF_1 +\cdots+ x^{t-1}F_{t-1}$ for $x \in \mathbf{N} \cup \{0\}$ and compute $S_i = F(i)$ for $i = 0,\ldots,n$. (Note that $S_0 = S$.)
3. Choose $f_1,\ldots, f_{t-1}$ uniformly at random from Z_q^*, construct a polynomial $f(x)=r+ f_1x +\cdots +f_{t-1}x^{t-1}$ for $x \in \mathbf{N} \cup \{0\}$ and compute $r_i = f(i)$ for $i = 0,\ldots,n$. (Note that $r_0 = r$.)
4. Send (S_i, r_i) to party Γ_i for $i = 1,\ldots, n$ secretly. Broadcast and $\delta_j = Comm(F_j, f_j)$ for $j = 1,\ldots, t-1$.
5. Each party Γ_i then checks whether its share (S_i, r_i) is valid by computing

$$Comm(S_i, r_i)= \prod_{j=0}^{t-1} \delta_j^{i^j} \quad . \qquad (2)$$

Distributed Key Generation Protocol Based on the Bilinear Pairing. We are now ready to construct a distributed key generation protocol, whereby a number of parties *without* a dealer can jointly generate a secret $K \in \mathcal{G}^*$ and its corresponding public value $\gamma = \hat{e}(K, P)$. We call this protocol "Distributed Key generation Protocol Based on the Bilinear Pairing (**DKPBP**)". Notice that the aim of **DKPBP** is analogous to Gennaro et al.'s [9] distributed key generation protocol for discrete-logarithm based cryptosystem in which a number of parties can generate a secret $k \in Z_q$ and its corresponding public value $g^k \in Z_p^*$ jointly, where g is a generator of Z_p^* and the primes p and q satisfy $q|p-1$.

To build up **DKPBP**, we need a distributed-version of the **UVSSBP** scheme, which we call "Distributed Unconditionally-secure Verifiable secret-sharing scheme based on the Bilinear Pairing (**DUVSSBP**)". With this protocol, a secret $S \in \mathcal{G}^*$ can be generated jointly by participating parties without a dealer. Due to lack of space, we refer readers to [2] for the details of **DKPBP**.

5. Our ID-based Threshold Signature Scheme

Description of the Scheme. Combining the various building blocks presented in the previous section and employing Hess' ID-based signature scheme [11] as a base scheme, we now construct the ID-based threshold signature scheme based on the Bilinear pairing, which we call "**IDTHSBP**" as follows. (For simplicity, we omit the details of sub-algorithms **CVSSBP** and **DKPBP**, and only describe the significant values resulted from them.)

• A common parameter algorithm GC(k): The PKG runs this algorithm to generate its private/public key pair and all the necessary common parameters. Precisely, the PKG performs the following.
- Choose a group $\mathcal{G}$ of prime order q and its generator P. Specify the Bilinear pairing $\hat{e}$: $\mathcal{G}\times\mathcal{G}\rightarrow\mathcal{F}$.
- Pick a master key s uniformly at random from Z_p^* and compute $P_{pub}= sP$.
- Choose two hash functions H_1: $\{0,1\}^* \rightarrow \mathcal{G}$ and H_2: $\{0,1\}^* \times \mathcal{F} \rightarrow Z_p^*$.
- Keep the master key s as secret and return the common parameter $cp = (\mathcal{G}, q, P, \hat{e}, P_{pub}, H_1, H_2)$.

• A private key extraction algorithm EX(cp, ID): On receiving a private key extraction query ID from any user, the PKG performs the following.
- Compute $Q_{ID} = H_1$ (ID) and $D_{ID}= sQ_{ID}$.
- Return D_{ID}.

• A private key distribution algorithm DK(cp, D_{ID}, t, n): A trusted dealer (as discussed in Section 1, this user could be the PKG itself) who possesses a private key D_{ID} associated with an identity ID performs the following.
- Run **CVSSBP** taking $(\mathcal{G}, q, P, \hat{e}, P_{pub}, H_1, t, n, D_{ID})$ as input to share D_{ID} among n signature generation servers, denoted by $\Gamma_1,\ldots, \Gamma_n$. By D^i_{ID} for $i = 1,\ldots, n$, denote each of the private key share of D_{ID} held by Γ_i. By α_k for for $k = 0,\ldots,t-1$, where t is a threshold, denote the public verification information output at the end of the execution of **CVSSBP**.

• A signature generation algorithm S(cp, D^i_{ID}, M) where $1 \le i \le n$: Each signature generation server Γ_i performs the following to jointly generate a signature on a given message M.
- Run **DKPBP** taking $(\mathcal{G}, P, t, n)$ as input to jointly generate a secret value K and a public value $\gamma =\hat{e}(K, P)$.
*Denote the resulting share of the sever Γ_i by K_i, for $i = 1,\ldots, n$. By $\beta_k= \prod_{i\in QUAL} \beta_{ik}$ for $k = 0,\ldots,t-1$, denote the public verification information output at the end of the execution of **DKPBP**. (Note that $\beta_0 = \hat{e}(K, P) = \gamma$. Note also that β_i's are the public values output from **DKPBP**. See [2] for more details.)
- Compute $v=H_2(M, \gamma)$.
- Broadcast $U_i= vD^i_{ID} +K_i$. (If Γ_i does not broadcast a value, we set U_i to *null*.)
- For server Γ_i where $i \in QUAL$, verify that

$$\hat{e}(U_i, P) = \left(\prod_{k=0}^{t-1} \alpha_k^{i^k}\right)^v \prod_{k=0}^{t-1} \beta_k^{i^k} \qquad (3)$$

- Construct U by computing $U = \sum \pi_i U_i$ where π_i denotes the Lagrange coefficient for the set Ψ such that $|\Psi| \ge t$.
- Return $\sigma = (U, v)$.

• A verification algorithm V(cp, ID, M, σ): Any user who wants to verify a signature σ = (U, v) on a message M performs the following.
- Compute $\gamma = \hat{e}(U, P)\hat{e}(Q_{ID},-P_{pub})^v$, where $Q_{ID} =H_1$(ID) .

- If $H_1(M, \gamma) = v$ then return "*Accept*", otherwise return "*Reject*".

Remarks on Design. We remark that although DK uses **CVSSBP** whose security is based on the mGBI problem, the security of DK is not relative to the mGBI problem but the CDH problem since the values P_{pub} and Q_{ID} are given as additional inputs.

Also, we remark that although the validity of the shares of D_{ID} and K are checked during the executions of **CVSSBP** and **DKPBP**, whether the partial signatures on the message M do reconstruct the original signature is not ensured. To resolve this problem, we have adopted Cerecedo et al. [5]'s technique in which the publicly available values output by **CVSSBP** and **DKPBP** are aggregated and the partial signatures are checked against them as presented in equation (3).

Providing Non-Repudiation. One criticism of ID-based signature schemes is that "non-repudiation", which is a very important property that signature schemes should possess, is not provided in the ID-based signature schemes due to the fact that the PKG always knows the user's private key and is capable of sign any messages at will. As discussed in [4, 11], the problem can be settled by distributing the PKG's master key into a number of multiple PKGs. Our scheme **IDTHSBP** can also be incorporated with this technique as follows. First, the master key s is jointly generated by the multiple PKGs using the technique of [14]. Holding a share s_i of s, each of the multiple PKGs then responds to the trusted user who is supposed to run the private key distribution algorithm DK of **IDTHSBP**'s private key extraction query with $D^i_{ID} = s_i Q_{ID}$ then the user collects these shares and recovers the private key D_{ID}. Having recovered D_{ID}, the user can run DK.

Security Analysis. It is proven in [2] that **IDTHSBP** is secure in the UF-IDTHS-CMA (Unforgeability of ID-based Threshold Signature against Chosen Message Attack) sense in the random oracle model assuming the CDH problem is computationally intractable. (Note that the UF-IDTHS-CMA notion which is an adaptation of the "unforgeability of a signature scheme against chosen message attack [10]" notion to the ID-based threshold signature setting is precisely defined in [2].)

6. References

[1] J. Baek and, Y. Zheng, "Identity-Based Threshold Decryption", to appear in Proc. of PKC 2004.

[2] J. Baek and, Y. Zheng, "Identity-Based Threshold Signature Scheme from the Bilinear Pairings", full version, available at http://phd.netcomp.monash.edu.au/joonsang.

[3] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", Proc. of the First ACM CCCS, pp 62-73, ACM Press, 1993.

[4] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", Proc. of CRYPTO 2001, LNCS 2139, pp 213-229, Springer-Verlag, 2001.

[5] M. Cerecedo, M. Matsumoto and H. Imai, "Efficient and Secure Multiparty Generation of Digital Signatures Based on Discrete Logarithms", IEICE Trans. Fundamentals., Vol. E76-A, pp 532-545,1IEICE, 1993.

[6] J. Cha and J. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Groups", Proc. of PKC 2003, LNCS 2567, pp 18-30, Springer-Verlag, 2003.

[7] P. Feldman, "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", Proc. of the FOCS, pp 427-437, IEEE, 1987.

[8] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Robust Threshold DSS Signatures", Proc. of EUROCRYPT '96, LNCS 1070, pp 354-371, Springer-Verlag, 1996.

[9] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Secure Distributed Key Generation for Discrete-Log Based Cryptosystem", Proc. of EUROCRYPT '99, LNCS 1592, pp 295-310, Springer-Verlag, 1999.

[10] S. Goldwasser, S. Micali and R. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks", SIAM Journal on Computing, 17, 2, pp 281-308, 1988.

[11] F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings", Proc. of SAC 2002, LNCS 2595, pp 310-324, Springer-Verlag, 2002.

[12] A. Joux, "The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems", Proc. of ANTS-V, LNCS 2369, pp 20-32, Springer-Verlag, 2002.

[13] T. P. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing", Proc. of EUROCRYPT '91, LNCS 576, pp 129-140, Springer-Verlag,1992.

[14] A. Shamir, "How to Share a Secret", Communications of the ACM, Vol. 22, pp 612—613, 1979.

[15] A. Shamir, "Identity-based Cryptosystems and Signature Schemes", Proc. of CRYPTO '84, LNCS 196, pp 47-53, Springer-Verlag, 1984.