On the Security of Encrypted Secret Sharing

Johannes Braun
TU Darmstadt
jbraun@cdc.informatik.tu-darmstadt.de

Alexander Wiesmaier
AGT International
awiesmaier@agtinternational.com

Johannes Buchmann
TU Darmstadt
buchmann@cdc.informatik.tu-darmstadt.de

Abstract

Sensitive electronic data must often be kept confidential over very long periods of time. Known solutions such as conventional encryption, cascaded encryption, and information theoretic schemes suffer from several weaknesses or serious disadvantages that we shortly discuss. We present a method for combining arbitrary encryption algorithms. The scheme has the following properties: (1) It is a (k, n)-threshold scheme, i.e. only $k \leq n$ of the n applied algorithms are needed for decryption. (2) The scheme's effective bit security is the sum of the lengths of the k shortest keys. (3) Under adaptive chosen plaintext attacks, this security level remains intact until at least k algorithms are compromised. (4) Under adaptive chosen chipertext attacks, the security level decreases with each compromised algorithm at most by the corresponding key length. (5) The scheme increases the effective key lengths of repeatedly applied encryption algorithms.

1 Introduction

1.1 Motivation

Certain electronic data has to be confidential for very long periods of time. Examples are governmental or military secrets and electronic healthcare data that may have to be kept confidential during the entire lifetime of the person it belongs to or even longer (e.g. in case of hereditary diseases). The problem is that all practical encryption algorithms known today are not suitable for this application. Their security declines over time due to improvements in cryptanalysis and availability of resources. Furthermore, the algorithms might be broken without premonition.

Once a ciphertext is known by potential attackers, the encryption, in contrast to digital signatures, cannot be prolonged in case the used algorithm or key is about to become insecure. An adversary once holding a cryptogram can store it and wait until the encryption algorithm becomes weak or broken. Thus, a complete and guaranteed removal of the cryptogram on behalf of the owner is not achievable once it has left the owner controlled environment. The possibility of encryption renewal theoretically exists in case the cryptogram has not been copied by any potential adversary at the time of renewal and if it is possible to remove it unrecoverably. However, this requirements are only met in very rare and special circumstances.

Thus, there is a need for practical ways to protect confidential data over very long periods of time. Ideally, the protection methods should not be threatened by advances in cryptanalysis or the emergence of new powerful resources.

1.2 Contributions

We generalize Schneier's OTP based scheme of combining block ciphers (OTPCC, cf. §1.3) to a system working with arbitrary perfect (k, n)-threshold secret sharing schemes (cf. §2.1). The resulting system, that we call General Secret Sharing based Cipher Combining (GSSCC), is a (k, n)-threshold system where only $k \leq n$ of the n combined ciphers are needed for decryption.

We prove that combining different ciphers is not a necessary condition for security amplification, but that reusing the same cipher with independent keys is sufficient, if some additional requirements are met.

We give tight security estimates for GSSCC, thereby improving the accuracy of the boundaries given for OTPCC so far. The effective bit security of GSSCC is given by the sum of the k shortest keys. The protection against adaptive chosen plaintext attacks (CPA2) stays on this level until at least k ciphers are compromised if GSSCC is applied in a special block mode (cf. §7). The protection against adaptive chosen chipertext attacks (CCA2) decreases at most by the corresponding key length for each compromised cipher.[1]

First reflections on information theoretic results indicate that GSSCC might conserve a certain security level even if all algorithms are compromised. This is to be investigated further.

1.3 Related Work

The only encryption algorithm known today that is provably secure and guarantees perfect confidentiality is the *One-Time-Pad* (OTP) invented by Vernam [34] in 1926.

[1] Note that standard mechanisms to protect from CCA2, such as electronic signatures, can be used with GSSCC.

The information theoretic or perfect security of this scheme was proven in 1949 by C. E. Shannon [29].

The OTP is not practicable for data storage. In order to ensure perfect security it is necessary to use an encryption key that is at least of the same size as the message to be encrypted [29]. Additionally, the key must not be reused and has to be generated uniformly at random. That means the use of this encryption scheme results into the same amount of key data as the amount of data to be encrypted. The keys have to be stored confidentially and therefore there is no benefit anymore, as the problem of securely storing the data is only deferred to securely storing the key.

Regarding data transport with OTP, the situation is similar. In order to realize this, special key exchange mechanisms are necessary. There are a few proposals for such systems, that all have their own practicability problems. See e.g. [4] for a survey on such schemes.

Another method to store data in a way that guarantees information theoretic security is the use of *Perfect Secret Sharing Schemes* (PSS) [30] (cf. §2.1). Such schemes are not based on encryption but divide the secret into so called *shares* in a way that the possession of less than k shares reveals absolute no information about the secret [28]. The parameter k can be chosen by the user. If shares are compromised, they have to be made invalid before a total of k shares are compromised. There exist solutions to this problem [15, 32]. In general, they are based on share renewal. However, the fundamental condition is that after this step all non compromised shares must not be available anymore. In case this cannot be guaranteed, an adversary might get hold of old shares and use them to reconstruct the secret with the shares compromised before. That again implicates that a secure and complete removal of shares has to be possible. But this is not always guaranteed in distributed storage. Additionally, there is the need for some kind of trust to the share holders not to cooperate illicitly. And there is another problem. If secret sharing is used to achieve long term security, one faces the problem of securely transferring the shares to the share holders.

In [29] Shannon proves some very strong secrecy properties of bijection families if certain requirements are met. In simple words and related to our scope he shows that, if encrypting uniformly distributed random numbers, an attacker has no other chance than guessing the key, independent from the attackers's power and the amount of intercepted ciphertext. For this, only the inherent bijective properties of the cipher (and not its encryption strength) is relevant. The problem here is to guarantee that the necessary requirements are met. Confer §2.2 for more details.

A common way to generate stronger ciphers from weaker ones is called *Multiple Encryption*. It means encrypting a plaintext multiple times using the same algorithm with different keys. There are many ways of using multiple encryption. There is double or triple encryption with two or three keys and in different modes, just to name a few. For an overview see [27]. A well known application of multiple encryption is 3DES [25]. In fact, multiple encryption may increase the key length if the algorithm does not form a group [27], which was proven for DES [7]. However, using n different and independent keys does not necessarily lead to a key length increase of factor n. In [12] Gaži and Maurer show that $n = 3$ is the smallest n providing a substantial improvement over single encryption. The security increase reached by $n > 3$ is left as an open question.[2] Meet-in-the-middle-attacks [23] push the effective key length increase of triple encryption significantly below the factor three. This shows that it is not an easy task to define the security level of multiple encryption schemes.

Cascading is similar to multiple encryption but uses different encryption algorithms. The use of cascading techniques has similar disadvantages as multiple encryption. There is no guarantee that combining different algorithms increases the security. However, when using independent keys for the cascading, proofs exist that cascading is at least as secure as the first algorithm in the cascade [22] or, if the algorithms commute[3], is at least as strong as the strongest algorithm in the cascade [11, 27].

In [27] Schneier proposes the above mentioned OTPCC scheme for combining block ciphers. It works as follows:
1. Share the secret using the OTP (see §2.1).
2. Encrypt each share using another cipher and independent keys chosen uniformly at random.

From PSS follows that the knowledge of $k - 1$ shares does not reveal anything about the secret. As long as OTP together with the applied ciphers has no homomorphic properties (cf. §5), it is clear that each algorithm has to be broken to reveal the secret. Thus, the scheme is guaranteed to be at least as secure as all applied not compromised ciphers.

2 Background

2.1 Perfect Secret Sharing

Perfect secret sharing schemes provide information theoretic security and were first invented independently by Shamir [28] and Blakley [3] in 1979. While Shamir uses the characteristics of polynomials, Blakely makes

[2] The work actually deals with cascading ideal ciphers, which is in this case equivalent to multiple encryption.

[3] Two ciphers C, Q commute if $C_i(Q_j) = Q_l(C_m)$ for every i, j and corresponding l, m [29].

use of the intersection of non parallel hyperplanes. It is a well known fact that to achieve information theoretic security, each share has to be at least as large as the secret itself [20]. Concerning threshold schemes, we focus on Shamir's secret sharing scheme (SSSS). It is an ideal scheme [5, 10], as the shares have exactly the same size as the secret, which obviously is optimal for a perfect secret sharing scheme. Additionally, we consider a simple straight forward secret sharing construction based on OTP.

We begin by giving a formal definition of (k, n)-secret sharing schemes with parameters $n, k \in \mathbb{N}, k \leq n$. Let $|A|$ denote the cardinality of the set A. We denote random variables with upper case letters with a hat e.g. $\hat{M}$, their domain with upper case letters, e.g, M and elements of a domain $m \in M$ with lower case letters.

In the following let $\hat{M},\ \hat{Y}_i$ for $i = 1, .., n$ and $\hat{Y}(m)$ be random variables with their domains $M,\ Y_i$ for $i = 1, .., n$ and $Y(m)$. M denotes the set of all possible secret messages and Y_i is the set of all possible shares at index i. $Y(m)$ is the set of all possible valid sharesets of cardinality n for message $m \in M$, where a shareset is of the form $(y_1, ..., y_n),\ y_i \in Y_i$. We define $Y = \bigcup_{m \in M} Y(m)$ as the set of all valid sharesets of cardinality n for any message $m \in M$.

Additionally, we define the subset operator "$\subseteq$" on sharesets a, b such that $a = (y_{j_1}, ..., y_{j_k}) \subseteq (y_1, ..., y_n) = b$ iff $\{j_1, ...j_k\} \subseteq \{1, ..., n\}$ and we say two subsets a, b are equivalent $a \sim b$ iff $(|a|, |b| \geq k) \wedge (a \subseteq c) \wedge (b \subseteq c),\ c \in Y$. $|a|$ denotes the cardinality namely the number of shares contained in the shareset a.

In a (k, n)-secret sharing scheme on input $m \in M$ and parameters n, k a set of so called shares $(y_1, ..., y_n),\ y_i \in Y_i$ is randomly generated with respect to the property that m can be recovered from any subset of shares of cardinality at least k. We call a set with that property a *valid shareset*. If less than k shares do not provide any information about the secret, the scheme is called perfect. In case $k < n$ the scheme is called a (k, n)-threshold scheme.

Definition 1 ((k,n)-Secret Sharing Scheme (SSS)) *A secret sharing scheme $\mathcal{S}$ is a tuple $(M, Y, \mathsf{Share}, \mathsf{Recover})$ with the following properties:*

- *M denotes the set of all possible messages*
- *Y denotes the set of all possible valid sharesets*
- $\mathsf{Share}(m, n, k)$ *on input $m \in M$ and parameters $n, k \in \mathbb{N},\ k \leq n$ outputs a randomly selected set of shares $y(m) = (y_1, ..., y_n) \in Y(m)$ for the secret message m.*
- $\mathsf{Recover}(y_{j_1}, ..., y_{j_k})$ *outputs m' with $m' = m$ iff $(y_{j_1}, ..., y_{j_k}) \subseteq y(m),\ y(m) \in Y(m)$*

Then, perfect secret sharing is defined as follows.

Definition 2 (Perfect Secret Sharing (PSS)) *A (k,n)-secret sharing scheme $\mathcal{S} = (M, Y, \mathsf{Share}, \mathsf{Recover})$ is called perfect iff $\forall y \in Y$ and $\hat{A} = (\hat{Y}_{j_1}, ..., \hat{Y}_{j_v}) \subseteq (\hat{Y}_1, ..., \hat{Y}_n)$, $a = (y_{j_1}, ..., y_{j_v}) \subseteq y$ holds:*

- *$\forall \{j_1, ...j_v\} \subseteq \{1, ..., n\}$ with $v \geq k$ there is a unique $m \in M$ such that $Pr(\hat{M} = m | \hat{A} = a) = 1$ and*
- *$\forall \{j_1, ...j_v\} \subseteq \{1, ..., n\}$ with $v < k$, $\forall m \in M$: $Pr(\hat{M} = m | \hat{A} = a) = Pr(\hat{M} = m)$*

The OTP based perfect secret sharing scheme is a trivial construction for a (k, k)-scheme. In that case $k - 1$ bitstrings of the same length as the secret are chosen uniformly distributed at random and form the first $k - 1$ shares. The kth share is the binary addition (XOR) of that shares and the secret. Clearly, the binary addition of any subset of at most $k - 1$ shares is an OTP encryption of the secret and therefore information theoretically secure.

As stated above, Shamir's secret sharing scheme (SSSS) [28] makes use of the characteristics of polynomials over finite fields. To share a secret with a (k, n)-threshold, $k - 1$ secret coefficients $a_i \in F,\ i = 1, 2, \ldots, k - 1$, where F is a finite field, are chosen uniformly distributed at random. The secret $s \in F$ forms the constant term of the polynomial $f(x) = s + \sum_{i=1}^{k-1} a_i x^i$. For n mutually different $x_j \in F,\ j = 1, 2, \ldots, n$, the evaluations of the polynomial $y_j = f(x_j)$ form the shares.[4] From each subset of k shares and the corresponding x_j the secret can unambiguously be reconstructed by interpolation using Lagrange's formula $s = f(0) = \sum_{i=1}^{k} y_i \prod_{l=1, l \neq i}^{k} \frac{x_l}{x_l - x_i}$. For SSSS the following theorem can be shown [28, 33] and, by construction, also holds for e.g. the OTP based secret sharing.

Theorem 1 *Let $y(m) = (y_1, .., y_n) \in Y(m)$ (instantiated as SSSS or OTP based SS). Then the shares in any subset $a \subset y(m)$ with $|a| < k$ are uniformly distributed and mutually independent.*

Using Shamir's method to share arbitrary secrets requires arbitrarily large fields, large number arithmetic and inefficiencies. Miyamoto et al. [24] provide an approach avoiding large fields and admitting an efficient implementation. The approach is to first split the secret into blocks such that each block is within a given field. Then each block is shared individually using SSSS while reusing the x_j. After sharing all blocks, all shares belonging to the same x_j are assembled respectively and build vectors of shares.

[4] Note that it is also possible to choose any subset of $k - 1$ shares at random (as in the OTP based SS), use these shares and the secret s to interpolate the polynomial and compute the missing $n - k + 1$ shares as evaluations of that polynomial.

2.2 Ideal Secrecy Systems

We explain *ideal secrecy systems* [27, 29] as a basis for later security considerations. We assume the reader to be familiar with Shannon's definition of entropy [29]. Let M denote the set of possible plaintexts or messages, and K denote the set of possible keys. Furthermore $m \xleftarrow{\$} M$ denotes that m is drawn uniformly at random from M.

Shannon defines ideal secrecy systems as such systems where any ciphertext only analysis (even exhaustive key search) gives many equiprobable decryptions [13] and therewith many equiprobable candidates for the actually applied key, independent on how much ciphertext is intercepted. He defines the two measures equivocation of message, denoted with $H_E(M)$, and equivocation of key, denoted with $H_E(K)$, for the average number of reasonable decryptions and keys respectively, depending on the amount of intercepted ciphertexts. Note that $H_E(K) \geq H_E(M)$ [29], as a ciphertext may decrypt to the same plaintext for different keys.

Strongly ideal secrecy systems are such systems where $H_E(K)$ stays constant at its initial value $H(K)$, the entropy of the cryptosystem. $H(K)$ is a measure of the size of the keyspace. If $k \xleftarrow{\$} K$ then [27]: $H(K) = log_2|K|$. In simple words, given a strongly ideal secrecy system, even an exhaustive key search leaves each $k \in K$ equiprobable independent from the amount of given ciphertext. Thus, an adversary cannot do better than choosing one key at random from the entire keyspace. Note that given one or more plaintext-ciphertext pairs, this may reveal a unique key. See [29] for formal definitions. Additionally, Shannon uses the term of a closed cipher. That is a cipher where for each possible message and for each different key there is exactly one cryptogram and vice versa.

Definition 3 (Cipher) *A cipher $\mathcal{C} = (P, C, K, \mathsf{E}, \mathsf{D})$ consists of the sets P, C, K and a tuple of algorithms E and D:*

- *P denotes the set of all possible plaintexts*
- *C denotes the set of all possible ciphertexts*
- *K denotes the set of all possible keys*
- *$\mathsf{E}_k(m)$ on input $m \in P$ and key $k \in K$, outputs the encryption $c \in C$.*
- *$\mathsf{D}_k(c)$ on input $c \in C$ and key $k \in K$ outputs the decryption $m \in P$.*
- *$\forall k \in K$, $m \in P$ it holds that $D_k(E_k(m)) = m$*

Definition 4 (Closed Cipher) *A cipher $\mathcal{C} = (P, C, K, \mathsf{E}, \mathsf{D})$ is closed if $\forall k \in K, p \in P : \mathsf{E}_k(p) \in C \ \wedge \ \forall k \in K, c \in C : \mathsf{D}_k(c) \in P$*

With this Shannon proves the following theorem.

Theorem 2 *If $\mathcal{C}$ is closed and each $p \in P$ appears with the same probability, then $\mathcal{C}$ is strongly ideal.*

It is also known that in general, given any natural language the ideal characteristic can be approximated. The use of compression to reduce or eliminate redundancy is a natural approach [27]. However, there are several disadvantages. Ideal systems rapidly become complex and have a bad error propagation characteristic. The system must be closely matched to the language, requiring extensive studies of the respective language. Even small changes or errors in the statistical structure make such schemes vulnerable to analysis. It is not always possible to achieve ideal secrecy with a system of finite complexity [29] and furthermore one given plaintext-ciphertext pair may reveal the key, thus undermining ideal secrecy.

In the following we show how the ideal secrecy can be exploited for arbitrary languages without the need for any knowledge about the characteristics and highly complex transformations of the secret to be encrypted. Additionally, the presented combiner prevents the leakage of plaintext-ciphertext pairs by construction.

3 GSSCC - General secret sharing based cipher combining

For our construction we apply the idea described by Schneier of combining several block ciphers using the OTP and generalize it to arbitrary perfect secret sharing schemes including threshold schemes.

First, we describe the encryption and decryption of the General Secret Sharing Cipher Combining scheme. Let $\mathcal{C}_i = \{P_i, C_i, K_i, \mathsf{E}_i, \mathsf{D}_i\}$ be closed ciphers according to Definition 4. Let further $\mathcal{S} = (M, Y, \mathsf{Share}, \mathsf{Recover})$ be any perfect secret sharing scheme according to Definition 2. Then $\mathcal{GSSCC}$ is defined as follows.

Definition 5 (GSSCC) *General secret sharing cipher combining $\mathcal{GSSCC}$ is a tuple*
$(M, \Gamma, \Gamma^*, \Pi, \mathcal{S}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec})$ *with the following properties:*

- *M denotes the set of all possible messages*
- *Γ denotes the set of all possible ciphertexts, $\Gamma = C_1 \times ... \times C_n$, with C_i the ciphertext space of the cipher $\mathcal{C}_i$*
- *Γ^* denotes the set of all possible k-subsets of ciphertexts, $\Gamma^* = \{C_{j_1} \times ... \times C_{j_k} | \{j_1, ..., j_k\} \subseteq \{1, ..., n\}\}$, with C_i the ciphertext space of the cipher $\mathcal{C}_i$*
- *Π denotes the set of all possible keys, $\Pi = K_1 \times ... \times K_n$, with K_i the key space of the cipher $\mathcal{C}_i$*
- *$\mathcal{S}$ denotes the applied perfect secret sharing scheme*
- *$\mathcal{C}$ denotes the set of ciphers $\mathcal{C}_i$, $i \in \{1, ..., n\}$*
- *$\mathsf{Enc}_\pi(m, n, k, \mathcal{S}, \mathcal{C})$ on input $m \in M$ and parameters $n, k \in \mathbb{N}$, $k \leq n$, perfect secret sharing scheme $\mathcal{S}$, a set of ciphers $\mathcal{C}$ and key $\pi \in \Pi$ outputs a ciphertext $c(m) \in \Gamma$ for the message m.*
- *$\mathsf{Dec}_\pi(c(m)^*, \mathcal{S}, \mathcal{C})$ on input of a partial ciphertext $c(m)^* \in \Gamma^*$, π, $\mathcal{S}$, $\mathcal{C}$ outputs $m' \in M$ with $m' = m$ if $c(m)^* \subseteq c(m)$*

Note that for an instantiation of GSSCC $n, k, \mathcal{S}$ and $\mathcal{C}$ are fix, thus from now we write $\mathsf{Enc}_\pi(m)$ and $\mathsf{Dec}_\pi(c(m)^*)$. The Encryption Enc: $M \overset{share}{\rightarrow} Y \overset{encrypt}{\rightarrow} \Gamma$ uses the algorithm Share of $\mathcal{S}$ and the encryption algorithms E_i of $\mathcal{C}_i$ with $\mathcal{C}_i \in \mathcal{C}$ as subroutines and works as follows:

1. $y(m) = (y_1, ..., y_n) = \mathsf{Share}(m, n, k)$
2. $c(m) = (c_1, ..., c_n) = (\mathsf{E}_{1,k_1}(y_1), ..., \mathsf{E}_{n,k_n}(y_n))$

The decryption Dec: $\Gamma^* \overset{decrypt}{\rightarrow} Y^* \overset{recover}{\rightarrow} M$ (note that $Y^* = \{Y_{j_1} \times ... \times Y_{j_k} | \{j_1, ..., j_k\} \subseteq \{1, ..., n\}\}$) uses the algorithm $\mathsf{Recover}$ of $\mathcal{S}$ and the decryption algorithms D_i of $\mathcal{C}_i$ with $\mathcal{C}_i \in \mathcal{C}$ as subroutines and for $\{j_1, ..., j_k\} \subseteq \{1, ..., n\}$ works as follows:

1. $y(m)^* = (\mathsf{D}_{j_1,k_{j_1}}(c_{j_1}), ..., \mathsf{D}_{j_k,k_{j_k}}(c_{j_k}))$
2. $m = \mathsf{Recover}(y_{j_1}, ..., y_{j_k})$

The correctness of the scheme follows from the correctness of the applied secret sharing scheme and ciphers.

4 Security in the ideal cipher model

In this section we analyze the security of the generic GSSCC construction in the ideal cipher model. Refer to §5 for considerations concerning a specific instantiation.

In the ideal cipher model ideal encryption schemes are assumed, i.e. that the applied ciphers act like random permutations [1]. It might be possible, that ideal ciphers do not exist [8, 9]. However, the security in the ideal cipher model shows, that it is necessary to exploit weaknesses of the primitives to break the scheme.

Due to the ideal cipher model, exhaustive key search is the only possible attack on the ciphers. For the sake of simplicity we consider all applied ciphers to have the same key space K and therewith the same key length $l = log_2(|K|)$ for the analysis. We provide the differences for the case of different key lengths afterwards.

We assume the applied keys to be drawn uniformly at random and independent and the ciphers to be closed. As we have ideal ciphers, we exclude collisions of decrypted values namely $\mathsf{D}_k(c) = \mathsf{D}_{k^*}(c)$ for $k \neq k^*$ and $c \in C$. The probability of such a collision is at most $1/2^{\frac{|K|^2}{|P|}}$ by the birthday bound. Therefore, such collisions do not significantly reduce the number of possible decryptions $\mathsf{D}_k(c) \in P,\ k \in K$ for $|P|$ significantly larger than $|K|^2$. E.g. $|P| = 2^{1024}$ and $|K| = 2^{256}$ results in a probability of a collision of $1/2^{513}$.

Furthermore, as each cipher is an ideal cipher with key length l and due to the property of perfect secret sharing, that each subset of k shares can be used to uniquely reconstruct the shared secret, an adversary gains no further exploitable information from additional shares. Therefore, wlog. we use $n = k$ for our analysis.

4.1 Exhaustive key search

First, we consider an adversary who only knows ciphertexts, i.e. sets of encrypted shares $c(m) = (c_1, ..., c_k)$. Therefore, the key for the scheme is of size $\tilde{l} = k * l$. The most naive attack is exhaustive key search choosing one key at a time, decrypting the shares and combining them, which requires $2^{\tilde{l}-1}$ attempts on average to find the key. In detail this means $2^{\tilde{l}-1} * k$ single decryptions, $2^{\tilde{l}-1}$ recover operations and $O(1)$ space.

To avoid repeated decryptions of a share with the same key, the adversary might set up a list of all possible decryptions for each share. This yields 2^l decryptions for each share and therewith $k * 2^l$ decryptions in total and $O(k * 2^l)$ space to store the lists. Therewith, there are $2^{\tilde{l}}$ possible combinations of shares taken from the lists as input for the recovery function of the secret sharing scheme, leading to an effort of $2^{\tilde{l}-1}$ recover operations on average for exhaustive search.[5]

Note that for $k < n$ and different key sizes $l_1, ..., l_n$ of the ciphers, the effective key length is reduced to the sum of the k shortest key lengths.

4.2 Meet-in-the-Middle attack

Given at least one plaintext-ciphertext pair for the GSSCC instantiation, namely a pair $(m, c(m))$ this can be exploited for a meet-in-the-middle (MitM) attack to reduce the computational effort in exhaustive search. If the secret sharing scheme allows for partial reconstruction,[6] a time-memory trade off is possible. Let $(y_1, ..., y_k)$ be a set of k shares, $a \subset (y_1, ..., y_k)$ and $b = (y_1, ..., y_k) \backslash a$, then partial reconstruction means that the shares within a and b can be combined separately obtaining m_1 and m_2 where m can be reconstructed from these partial results.[7] Obviously, it is also possible to split the set of shares into more than two subsets. We denote partial reconstruction with $\mathsf{partRecover}(y_{j_1}, ..., y_{j_u}),\ for\ \{j_1, ..., j_u\} \subset \{1, ..., k\}$. The attack works as follows (exemplary for even k and ciphers with key length l)[8]:

1. Split the set of encrypted shares into two subsets $g_1 = (c_1, ..., c_{k/2})$ and $g_2 = (c_{k/2+1}, ..., c_k)$. Thus,

[5] Note that the adversary in both scenarios has to decide which of the generated possible plaintexts are reasonable, leading to additional efforts for verification or the need of several ciphertexts, which is not considered here in detail.

[6] which is true for Blakeley's, Shamir's and OTP based secret sharing

[7] Partial reconstruction is the XOR of a subset of shares when OTP based SS is applied or the sum of a subset of shares multiplied with the according Lagrange multipliers for SSSS.

[8] An uneven k leads to groups of size $\lceil k/2 \rceil$ and $\lfloor k/2 \rfloor$ with the according runtime and space requirements of the attack. For different key lengths of the applied ciphers, the shares can be allocated to the groups such that the key is split approximately into halves.

half of the key material is involved with the first and half is involved with the second group.

2. Let $\sigma = (\sigma_1, ..., \sigma_{k/2}) \in \Sigma = K^{k/2}$. For all $\sigma \in \Sigma$ compute $y_\sigma = (\mathsf{D}_{1,\sigma_1}(c_1), ..., \mathsf{D}_{k/2,\sigma_{k/2}}(c_{k/2}))$ and $m_{1,\sigma} = \mathsf{partRecover}(y_\sigma)$. Store the pairs $(m_{1,\sigma}, \sigma)$ into a list L_1. For different keys, the result of the partial reconstruction might be the same. However, this only implies several key candidates that have to be verified.
   Then replace each list entry $m_{1,\sigma}$ by $m \bullet m_{1,\sigma}$. "$\bullet$" is chosen according to the respective operation of the applied SSS.[9]
3. Sequentially compute all possible partial reconstructions $m_{2,\sigma'},\ \sigma' \in \Sigma$ using the elements in g_2 as described for g_1 omitting the combination with m.
4. Check for a collision with the values in L_1, namely if $m_{2,\sigma'} \stackrel{?}{\in} L_1$. If a collision is found, the associated key in L_1 concatenated with the current key σ' is a candidate for the complete key. Candidates can be verified with additional plaintext-ciphertext pairs.

This attack uses $O(2^{\tilde{l}/2})$ time and $O(2^{\tilde{l}/2})$ space compared to $O(2^{\tilde{l}})$ time for trivial exhaustive search. This is the best generic attack known so far and we consider intersecting the key into halves to be the best achievable attack concerning computational effort. First, we note that intersecting the encrypted shares and therewith the associated key parts into two subsets of different size implies, that for either the first or the second subset, there exist $t \geq \tilde{l}/2 + l$ possible combinations implying directly $O(2^t)$ time to either build the list or to check for collisions. Furthermore, splitting into more than two groups seems to bear no advantage, since in each verification k shares have to be considered in parallel due to the perfect security (see Definition 2). This implies, that no partial verification is possible on groups of less than k shares, which means that partial results of several subsets first have to be combined to provide two partial reconstructions of larger subsets for the above verification method.

In §7 we present a chaining mode that prevents from such MitM attacks by construction.

5 Concrete instantiation

It is clear, that as long as at least $n - k + 1$ of the used encryption schemes are considered secure, the combined scheme is secure. An adversary must break at least k encryption schemes as, due to the properties of perfect secret sharing, at least k decrypted shares are necessary to derive any information about the secret [27]. However, the randomness and independence of shares beares quite more security as it prevents from attacks on single algorithms.

As seen in the analysis above, in the ideal cipher model GSSCC can be used as a combiner to increase the key length arbitrarily (depending on k) and therewith to build an encryption scheme to withstand enormous computational power. However, it is not clear if ideal ciphers exist. Therefore, we analyze GSSCC instantiated with Shamir's secret sharing scheme and contemporary block ciphers to show, that the construction itself defends against several vulnerabilities of the applied ciphers.

For SSSS as an instantiation of $\mathcal{S}$ it is $M = Y_i = F\ \forall i \in \{1, .., n\}$, where F is the finite field used in SSSS (cf. §2.1 and [28] for details). Choosing $F = GF(2^r)$ as in [26], SSSS works on arbitrary bitstrings $b \in \{0,1\}^r$ in a natural way, as there is a bijektive mapping from F to the bitstrings of length r. Wlog. we are only concerned with messages $m \in M = \{0,1\}^r$ in the following, as this can be achieved by padding of shorter or intersecting larger messages into blocks.

We apply contemporary block ciphers $\mathcal{C}_1, ..., \mathcal{C}_n$ such as AES and Twofish. For simplicity let each cipher have the same block length bs, the same key length l and key space K and let r be a multiple of the block length, i.e. $r = v * bs,\ v \in \mathbb{N}$. Thus, for the application of the block ciphers in any chaining mode, e.g. CBC-Mode, we have $P_i = \{0,1\}^r\ \forall i \in \{1, .., n\}$ in a natural way.

We assume that the decryptions $\mathsf{D}_{i,k}(c)$ for different $k \in K_i,\ i \in \{1, ..., n\}$ are uniformly distributed over $\{0,1\}^r$ and appear random, which we consider reasonable for actual block ciphers as in general they should provide high diffusion by design [19]. Thus, $\mathsf{D}_{i,k}(c) \neq \mathsf{D}_{i,k^*}(c),\ k \neq k^*$ in general by the birthday bound for r significantly larger than $log_2|K| = l$, e.g. $r = 2 * k * l$.

Note that the ciphers must not have homomorphic properties concerning the combination operation of the applied SSS. That is, it must not be possible to first combine the encrypted shares and then decrypt the result with a specific cipher and a suitable key resulting into the correct plaintext.[10] For different keys, contemporary block ciphers represent different permutations. We assume these to prevent from such homomorphic properties in general.

5.1 Partial ideal secrecy under CPA2 attacks

In the following we consider an adversary that has access to an encryption oracle and can obtain pairs $(m, c(m))$ for adaptively chosen $m \in M$. With *partial ideal secrecy* we denote the property that each cipher in any subset of

[9] "$\bullet$" denotes the bitwise XOR for OTP based SS, for SSSS it denotes subtraction within the applied field F.

[10] A trivial example for that is the instantiation of GSSCC with OTP based SSS and OTP encryption of the shares.

cardinality at most $k-1$ of the ciphers provides strong ideal secrecy as the ciphers are closed and the input is random and uniformly distributed. We consider only the k weakest ciphers here and show that the security levels given by the analysis in the ideal cipher model also hold for a specific instantiation.

At first we state, that an adversary is never able to obtain the input for single ciphers, i.e. he is not able to obtain the shares $(y_1, ..., y_k)$ as for $F = GF(2^r)$, there are $2^{r*(k-1)}$ equiprobable outputs of the perfect secret sharing scheme (cf. Theorem 1). Thus, CPA attacks on the applied ciphers are prevented by construction. Furthermore, the shares of different executions of GSSCC are independent by construction. Thus, the inputs to the ith cipher are random, uniformly distributed and mutually independent, yielding strong ideal secrecy. This implies that the shares from different executions of GSSCC cannot be utilized to attack the ciphers respectively.

As the shares within each subset of cardinality at most $k-1$ are random uniformly distributed and mutually independent (cf. Theorem 1) an adversary can only attack one cipher in dependence of $k-1$ others. Meaning, as long as he does not fix $k-1$ shares, his view on the kth share is uniformly distributed and thus, protected by the strong ideal secrecy. The ideal secrecy can only be undermined by the knowledge of any function of the input to the cipher.

We model the information an adversary can obtain from $k-1$ encrypted shares and the given plaintext m as the function $\mathsf{info}(c_1, ..., c_{i-1}, c_{i+1}, ..., c_k, m)$. Now the adversary might use $\mathsf{info}(c_1, ..., c_{i-1}, c_{i+1}, ..., c_k, m)$ as additional input to the attack. However, as less then k shares are involved we have ideal secrecy for each involved cipher and the equiprobability of the keys. This means that $\mathsf{info}(c_1, ..., c_{i-1}, c_{i+1}, ..., c_k, m)$ at best may embody $2^{l*(k-1)}$ partial reconstructions leading to the same amount of candidate values for the ith share. Due to the assumption of uniform distribution of the decryptions applying different keys, the partial reconstructions are also uniformly distributed. By the birthday bound for e.g. $r = 2*k*l$ the probability of a collision of partial reconstructions is $\frac{1}{2}\frac{(2^{(k-1)*l})^2}{2^{2*k*l}} = \frac{1}{2}\frac{1}{2^{2*l}}$ which is negligible. Thus we exclude such collisions from our considerations.

As we assume uniform distribution and pseudorandomness of the decryptions applying different keys to each cipher, it is not possible to narrow down the results of partial reconstructions to a certain range of the plaintext space. Therewith, the probability to have a reasonable assumption on the plaintext of the ith share and therewith a reasonable result applying the attack on the ith cipher is $1/2^{(k-1)*l}$.

From this considerations we follow that any attack on a single algorithm cannot decrease the security of GSSCC.

We leave it as an open problem whether efficient combined attacks on contemporary block ciphers[11] exist where only the corresponding ciphertexts $(c_1, ..., c_k)$ and a function $f(y_1, ..., y_k)$ on the plaintexts are known. Thereby f is such that there are $|M|^{k-1}$ equiprobable inputs leading to the same result of f.

Assumed such an attack, then it must include at least $k/2$ of the ciphers applied within GSSCC to be better than the generic MitM attack concerning computational effort. This is due to the number of partial reconstructions when applying all possible keys to the not attacked ciphers.

We follow that any known attack on single algorithms does not decrease the security of GSSCC. This is due to the partial ideal secrecy, and the uniform pseudorandom decryptions for different keys. The security level implied by the MitM attack holds as long as no combined attack on more than $k/2$ ciphers is found.

5.2 Chosen-Ciphertext-Attacks

As the reconstruction of a secret is deterministic, the partial ideal secrecy of GSSCC can be undermined if an adversary is able to obtain plaintexts for arbitrarily chosen ciphertexts. Given an adversary enabled to execute chosen (CCA) or adaptive chosen ciphertext attacks (CCA2) on GSSCC, he might obtain plaintext-ciphertext pairs where the ciphertexts are equal except for one of the encrypted shares. This can e.g. be done by choosing the ciphertexts as $(c_1, ..., c_{k-1}, c_k)$ and $(c_1, ..., c_{k-1}, c'_k)$. Comparing the resulting plaintexts he can analyze the difference between the plaintexts with respect to the fact, that the difference involves only one of the applied ciphers and the corresponding key. This might bear vulnerabilities to the key material of that cipher. Therefore, this kind of attack should be prevented. This is possible by the use of signatures [18]. As signatures can be renewed over time, it is suitable to use signatures that are currently secure and renew them in case it becomes necessary. However, without the use of CCA2 preventing measures, the scheme's security level decreases with each compromised cipher at most by the corresponding key length.

5.3 Block Length of GSSCC

If Shamir's secret sharing is applied to instantiate GSSCC, the scheme's block length is defined by the size of the elements of the field F. For $F = GF(2^r)$ GSSCC encrypts message blocks of r bits at a time. The same holds for decryption. The applied ciphers may have different block lengths. However, to enable reconstruction of a

[11] attacks that attack some ciphers in parallel

message block, a share of size r bit has to be decrypted completely.[12]

To encrypt messages of arbitrary length, smaller messages can be padded. Messages longer than r bits can be shared using the approach of Miyamoto et al. [24] (cf. §2.1). The resulting vector of shares can then be encrypted using the block ciphers in any chaining mode of operation.

Another method to encrypt messages of arbitrary length would be to apply GSSCC itself in some chaining mode of operation tailored to the structure of GSSCC. As an example we give the description of a CBC-mode for GSSCC.

CBC-Mode for GSSCC Let $m = m_1 m_2, ..., m_t \in M$ be a sequence of t blocks of size r and GSSCC instantiated with SSSS as described at the beginning of this section with key π. Let $IV \xleftarrow{\$} \{0,1\}^r$ be the initialization vector, then: $c_1 = (c_{1,1}, ..., c_{1,k}) = \mathsf{Enc}_\pi(m_1 \oplus IV)$, $c_j = (c_{j,1}, ..., c_{j,k}) = \mathsf{Enc}_\pi(m_j \oplus c_{j-1,1} \oplus ... \oplus c_{j-1,k})$, $2 \leq j \leq t$ to obtain the ciphertext $c_0, ..., c_t = IV, (c_{1,1}, ..., c_{1,k}), ..., (c_{t,1}, ..., c_{t,k})$.

To decrypt one applies the decryption with key π as follows: $m_1 = \mathsf{Dec}_\pi(c_1) \oplus IV$, $m_j = \mathsf{Dec}_\pi(c_j) \oplus c_{j-1,1} \oplus ... \oplus c_{j-1,k}$, $2 \leq j \leq t$

Note that in this mode of operation all shares are needed for decryption and not only a subset of k shares, thus $n = k$ should be chosen for instantiation.[13] In case $r > bs$ the ciphers used as subroutines of the GSSCC-encryption need to be used in any chaining mode of operation itself.

6 Discussion

6.1 On the choice of ciphers to instantiate GSSCC

We generalized Schneier's approach of combining k ciphers using OTP based secret sharing schemes to any perfect secret sharing scheme. The generalization to arbitrary (k,k)-PSSS is straight foreward, since due to the information theoretic security if less than k shares are known (cf. §2.1), the adversary gains no information unless he breaks all k encryption schemes.

In both cases, (k,k)- and (k,n)-schemes, increasing k leads to an increase of security. On the one hand the effective key size is increased, on the other hand the characteristics of the additional cipher have to be considered when attacking GSSCC. Even a weak cipher e.g. with small key size adds additional security. But note that for the security of (k,n)-schemes only the k weakest ciphers are relevant and therefore only increasing n does not increase security but may decrease it.

[12]Concerning an instantiation with OTP based SS, the block length of GSSCC only depends on the block length of the applied ciphers.

[13]a (k,n)-threshold can be achieved by repeating the encryption with all reasonable k combinations of n shares

For the purpose of increasing the key space it is clear that all shares of the same secret have to be encrypted with different, randomly independently and uniformly chosen keys. Up to now we considered the use of several different ciphers within GSSCC. However, it might be desirable to use GSSCC with only one or less than k different ciphers to just increase the security of that ciphers against exhaustive key search and known plaintext attacks or to obtain a cipher with increased block length.

However, if only one cipher is used, it seems more probable that homomorphic properties exist or an attack on a set of ciphertexts where only a function of the plaintexts is known (as stated as open problem in §5.1), due to possible specific properties of the respective cipher that can be exploited. Up to now no such attack is known to us for current block ciphers.

Given that contemporary block ciphers are good pseudorandom permutations, they prevent from homomorphic properties preserving the benefits from the partial ideal secrecy of GSSCC. However, unknown weaknesses potentially have greater effect if only one cipher is used for instantiation. Using different ciphers further strengthens the security by the combination of their dedicated strengths.

6.2 Long term security

In general, secret key cryptosystems are considered to withstand quantum computers [2]. There is Grover's algorithm [14], but its effect can be compensated by doubling the key sizes [2]. The keyspace of our system can be chosen arbitrarily large to anticipate increasing computational power of adversaries. Up to now security levels of 256 bit are far out of reach for any practical attacks and 512 bit or even more are considered to hold for the next decades [21]. GSSCC furthermore withstands attacks on single ciphers. Thus, attacks on GSSCC are far more complex than on single algorithms as they must consider several algorithms in parallel, and it is left as an open problem if such attacks even exist (cf. §5.1). Concerning the exclusion of CCA adversaries, this can be guaranteed for long term applications, since signatures can be renewed over time whenever they are about to become insecure.

6.3 Further Remarks

Utilizing (k,n)-secret sharing has advantages compared to the use of (k,k)-secret sharing schemes. (k,n)-threshold schemes provide failure resistance, load balancing possibilities and flexibility in the use of algorithms, as a secret may be reconstructed on devices that need not implement all but only a k-subset of the used algorithms.

Considering instantiations of GSSCC the above OTP based scheme is clearly very efficient concerning computing time for sharing and reconstruction. When a (k,k)-

sharing is about to be used, OTP based secret sharing may be preferred. However, to achieve a (k, n)-sharing, any (k, n)-threshold scheme e.g. as SSSS is the better choice concerning storage space, as the shares stay optimal in size (cf. §2.1). Furthermore, SSSS has the additional advantage to allow arbitrary block sizes for GSSCC (cf. §5) whereas the block size for the OTP based approach is determined by the ciphers.

The major disadvantage of GSSCC is that combining k ciphers implies that the ciphertext is at least k times the size of the plaintext, by the lower bound of the size of shares of perfect secret sharing schemes.

7 Meet-in-the-middle secure block mode

We propose a new chaining mode, especially designed for the use with GSSCC. The mode is called MSB. We derive it from the above CBC-mode and apply several changes to prevent from MitM attacks as described in §4.2.

Let $m = m_1 m_2, ..., m_t \in M$ be a sequence of t blocks of size r and GSSCC instantiated with SSSS and key π. Remind that y_i denotes the ith share output by the secret sharing scheme, namely $y_i = \mathsf{D}_{i,k_i}(c_i)$. Let $IV \xleftarrow{\$} \{0,1\}^r$ be the initialization vector, then:
$c_0 = (c_{0,1}, ..., c_{0,k}) = \mathsf{Enc}_\pi(IV)$, $c_1 = (c_{1,1}, ..., c_{1,k}) = \mathsf{Enc}_\pi(m_1 \oplus IV)$, $c_j = (c_{j,1}, ..., c_{j,k}) = \mathsf{Enc}_\pi(m_j \oplus y_{j-1,1} \oplus ... \oplus y_{j-1,k-1})$, $2 \leq j \leq t$ to obtain the ciphertext $c_0, ..., c_t = (c_{0,1}, ..., c_{0,k}), ..., (c_{t,1}, ..., c_{t,k})$. To decrypt one applies the decryption with key π as follows: $IV = \mathsf{Dec}_\pi(c_0)$, $m_1 = \mathsf{Dec}_\pi(c_1) \oplus IV$, $m_j = \mathsf{Dec}_\pi(c_j) \oplus y_{j-1,1} \oplus ... \oplus y_{j-1,k-1}$, $2 \leq j \leq t$

The main differences to the CBC-mode are, that the IV is not provided in plaintext. Additionally the chaining is done with the decryption of $k-1$ shares respectively. These shares are uniformly distributed and independent (Theorem 1). Thus, even for a known plaintext m, the input to the GSSCC executions is unknown to an adversary, preventing from the MitM attack. Thus, the scheme's effective bit security is the sum of the lengths of the k keys.

As IV is chosen uniformly at random this yields strong ideal secrecy. Additionally, without having decrypted the previous block (or $k-1$ shares of the previous block), the view on the input of the current block is uniformly distributed and random implying strong ideal secrecy for each block respectively.

8 Applications

In general, GSSCC can replace the encryption algorithms in any application. This might for example be indicated where confidentiality over long periods is required. Besides that, GSSCC can be easily plugged into archival and storage solutions which apply secret sharing such as e.g. POTSHARDS [31] or [16]. A key management module would have to be integrated to store and manage the keys, yet the other processes can remain untouched. The solution is of special interest for e.g. cloud storage solutions, where the data is not held in an owner controlled environment. Given an architecture such as described in [17], the data processor located within the enterprise network can apply GSSCC and locally manage the keys.

Going one step further and following the approach of Cachin et al. [6], the several shares can, after encryption, be sent to different single domain clouds (at different geographical locations), thereby realizing what they call the intercloud. As GSSCC can provide redundancy and is fully flexible in its parameters, the dependability goals from the intercloud approach can be fully exploited.

However, one disadvantage of storing encrypted data using strong encryption is, that processing directly on the stored data is impossible. Processing requires local decryption and recombination. Yet, as explained in §2.1, data blocks of arbitrary size can be shared and hence encrypted individually. Thus, e.g. in case of medical records, each customer file can be encrypted and thereby retrieved independently from other files. As the keys can remain the same for different data blocks, key management efforts do not grow and the granularity can be chosen as required by the respective application.

9 Conclusion and Future Work

We have seen, that perfect secret sharing is applicable to generate cipher combiners that merge the strengths of several algorithms and are further secure against CPA2 attacks on the algorithms. GSSCC can also be used to increase the effective key length of single ciphers. Especially the randomness and uniform distribution of $k-1$ shares together with Shannon's ideal secrecy systems provide strong security properties.

The presented GSSCC scheme together with the associated MSB mode results in a bit security that is the sum of the lengths of the k shortest keys of the underlying ciphers. Regarding CPA2 attacks, the security remains on this level until at least k underlying ciphers are compromised. There is indication that a substantial security level remains even if k or more ciphers are compromised due to the need for combined attacks. Regarding CCA2 attacks, the security decreases with each compromised algorithm at most by the corresponding key length. Conventional CCA2 countermeasures, such as signatures, are usable with the scheme.

An open problem is the possible existence of homomorphic properties of the different ciphers under the PSS scheme, which would allow for the existence of efficient combined cipher attacks on GSSCC. As contemporary

block ciphers represent for each key a different permutation, such homomorphic properties seem unlikely. Nevertheless, a tight investigation on this is to be conducted.

The main disadvantage of GSSCC is the data multiplication. Adding one cipher to the scheme means increasing the amount of data by the size of the secret. Yet, the scheme leaves space for improvement regarding the data multiplication rate and the efficiency. Clearly, the former influences the latter. A promising approach to minimize the data multiplication is the incorporation of information dispersal algorithms [26] together with or even instead of perfect secret sharing. Applying OTP for secret sharing or encryption purposes promises to significantly increase the efficiency, due to the unbeatable performance of the underlying XOR operation. Especially as OTP keys seem to be reusable for random and uniformly distributed inputs. How this is done best is subject to future investigation. Another even more general approach for combining ciphers is cascaded application of GSSCC, e.g. in a tree structure, which needs to be examined in detail.

References

[1] J. Bender, M. Fischlin, and D. Kügler. Security analysis of the pace key-agreement protocol. In *Proceedings of the 12th International Conference on Information Security*, ISC '09, pages 33–48, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] D. J. Bernstein. Introduction to post-quantum cryptography. In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 1–14. Springer Berlin Heidelberg, 2009.

[3] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, Monval, NJ, USA, 1979. AFIPS Press.

[4] J. Braun, J. Buchmann, C. Mullan, and A. Wiesmaier. Long term confidentiality: a survey. *Designs, Codes and Cryptography*, 2012. Accepted for publication, to appear. Preliminary version: Cryptology ePrint Archive, Report 2012/449.

[5] E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes (extended abstract). In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 278–285, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[6] C. Cachin, R. Haas, and M. Vukolić. Dependable storage in the intercloud. Research Report RZ 3783, IBM Research, Aug 2010.

[7] K. W. Campbell and M. J. Wiener. Des is not a group. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 512–520, London, UK, 1993. Springer-Verlag.

[8] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51:557–594, July 2004.

[9] J.-S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In D. Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-85174-5_1.

[10] E. Dawson and D. Donovan. The breadth of shamir's secret-sharing scheme. *Comput. Secur.*, 13(1):69–78, 1994.

[11] S. Even and O. Goldreich. On the power of cascade ciphers. *ACM Trans. Comput. Syst.*, 3:108–116, May 1985.

[12] P. Gaži and U. Maurer. Cascade encryption revisited. In *ASIACRYPT '09 Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*. Springer-Verlag Berlin, 2009.

[13] P. R. Geffe. Secrecy systems approximating perfect and ideal secrecy. In *Proceedings of the IEEE*, volume 53, pages 1229–1230. IEEE Journals, 1965.

[14] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *PHYS.REV.LETT.*, 79:325, 1997.

[15] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. pages 339–352. Springer-Verlag, 1995.

[16] D. Hühnlein, U. Korte, L. Langer, and A. Wiesmaier. A comprehensive reference architecture for trustworthy long-term archiving of sensitive data. In I. Press, editor, *3rd International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Dez 2009.

[17] S. Kamara and K. Lauter. Cryptographic cloud storage. In *Proceedings of the 14th international conference on Financial cryptograpy and data security*, FC'10, pages 136–149, Berlin, Heidelberg, 2010. Springer-Verlag.

[18] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[19] L. R. Knudsen. Contemporary block ciphers. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, pages 105–126, London, UK, 1999. Springer-Verlag.

[20] H. Krawczyk. Secret sharing made short. In *CRYPTO '93: Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, pages 136–146, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[21] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14:255–293, 2001. 10.1007/s00145-001-0009-4.

[22] U. M. Maurer and J. L. Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology*, 6:55–61, 1993.

[23] R. C. Merkle and M. E. Hellman. On the security of multiple encryption. *Commun. ACM*, 24:465–467, July 1981.

[24] T. Miyamoto, S. Doi, H. Nogawa, and S. Kumagai. Autonomous distributed secret sharing storage system. *Syst. Comput. Japan*, 37(6):55–63, 2006.

[25] National Institute of Standards and Technology. Data encryption standard (DES). FIPS Publication 46-3, October 1999.

[26] M. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, 36:335–348, 1989.

[27] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

[28] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[29] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal, Vol 28, pp. 656715*, Oktober 1949.

[30] D. R. Stinson. An explication of secret sharing schemes. *Des. Codes Cryptography*, 2(4):357–390, 1992.

[31] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti. Potshards -a secure, recoverable, long-term archival storage system. *Trans. Storage*, 5:5:1–5:35, June 2009.

[32] A. Subbiah and D. Blough. Practical share renewal for large amounts of data, 2005.

[33] M. Tompa and H. Woll. How to share a secret with cheaters. *J. Cryptol.*, 1(2):133–138, 1988.

[34] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *American Institute of Electrical Engineers, Vol XLV, pp. 109115*, 1926.