Hybrid Secret Key Escrow Mechanisms as Counters

Esa Hyytiä
Telecommunications Research Center Vienna (ftw.)
Donau-City Straße 1, 1220 Vienna, Austria

***Abstract*—Many of the cryptographic primitives can be used in several ways. One interesting application of the Shamir's secret sharing scheme in the context of privacy aware traffic monitoring is to escrow a secret key after m suspicious events have been observed [1]. In the proposed system, a so-called front-end component encrypts the monitored data traffic, which is then stored at the back-end. At the same time, the front-end analyzes the traffic, and if suspicious packets are observed, this is indicated to the back-end by revealing one share of the corresponding encryption key. Once m suspicious events have been detected, the back-end can disclose the secret key, decrypt the particular traffic flow, and carry out further investigations. In this paper we study the secret sharing scheme as a counter at the limit when the threshold m is relatively large. We first analyze how the scheme behaves as m approaches the maximum possible value of $\mathrm{p}-1$, where p is a prime number (design parameter). Then, we also analyze a probabilistic version developed to overcome the limited counting range, or excessive reporting overhead, by revealing shares only with a certain probability after each event, and provide expressions describing the resulting inaccuracy from the introduced randomness. Finally, we also propose a hybrid solution to mitigate the otherwise detoriating performance by using a forward error correction scheme similar to LT codes to encode the shared secret revealing process.**

***Index Terms*—Shamir's secret sharing, key escrow, LT codes.**

I. Introduction

Many coding and cryptographic schemes share similar mathematical constructions. Moreover, cryptographic primitives such as RSA can be used, e.g., to guarantee authentity by a digital signature, or to allow the public to send encrypted messages which only a certain party can decrypt. In other words, these schemes are very versatile and by creative use, they can be applied to a multitude of problems.

In an ideal case one can trust the involved parties, making the cryptographic techniques obsolete at the same time. In practice, this unfortunately is not the case, but instead such methods must be used to ensure that each party has access only to the data they are entitled to. In a recent paper [1], Bianchi et al., present an interesting approach for privacy-preserving traffic monitoring in data networks. The authors devise a scheme which efficiently combines data anonymization with anomaly detection by using several cryptographic primitives as building blocks. The aim is to prevent the higher layer party from identifying flows and users unless it is justified due to a suspicion of anomalous traffic. One important part of the system is the use of the Shamir's secret sharing mechanism to escrow a secret key to the higher layer after m anomalous events have been observed, i.e., each suspicious event triggers the system to publish one share. The secret key discloses the identity of the malicious flow and allows further investigations.

In the present paper we consider this particular usage of the Shamir's secret sharing algorithm to count events in such a way that the secret is disclosed only when a given threshold m is exceeded. In the limit of large m, the scheme has certain reliability problems due to a positive probability of accidentally revealing the same share twice. Our aim is to explore the possibilities to extend the counting range beyond the limits of the straightforward scheme by occasionally revealing also combined shares in analogy with the rateless LT codes [6].

The rest of the paper is organized as follows. Section II contains the necessary definitions. In Section III, the basic and thinning schemes are analyzed. In Section IV, we propose and analyze a hybrid extension that efficiently mitigates the reliablity problem for large m. Section V concludes the paper.

II. Preliminaries

In this section, we will provide the necessary background and notation for the later analysis.

A. Shamir's Secret Sharing Scheme

The task of secret sharing schemes is to distribute a certain secret to n parties in such a way that this information is only useful when a sufficient number, $m \leq n$, of the parties together decide to disclose the secret [2]. Such a system is called a (m, n)-threshold scheme, and is considered to be *secure* only if no combination of fewer than m shares reveals any extra information about the secret. One such elegant and secure scheme, proposed by Shamir in [3], is based on polynomial

$$P(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0, \quad (1)$$

where a_0 is the *shared secret*. Computations are carried out using modulo arithmetic in finite field defined by some large prime p, i.e., the field consists of the elements $\{0, 1, \ldots, p-1\}$. Thus, $0 \leq a_i < p$ and also $0 \leq x < p$. The secret sharing is achieved by distributing n pairs,

$$\{x_1, P(x_1)\},\ \{x_2, P(x_2)\},\ \ldots\ \{x_n, P(x_n)\},$$

to n different parties, where the $x_i \neq 0$, $x_i \neq x_j\ \forall\ i \neq j$, and $n \geq m$. As each pair $\{x_i, P(x_i)\}$ corresponds to a point in a curve $P(x)$, we refer to these bits of information as *points*. It follows that the knowledge of any m pairs out of n defines, in straightforward manner, the polynomial $P(x)$, and consequently, also the constant a_0, which plays the role of the shared secret. Furthermore, knowledge of any $m-1$ points gets one no where. Hence, this scheme provides the means for n parties to share a secret in such a way that any combination of m parties is capable of disclosing it, and that no combination of $m-1$ or less parties can derive the secret.

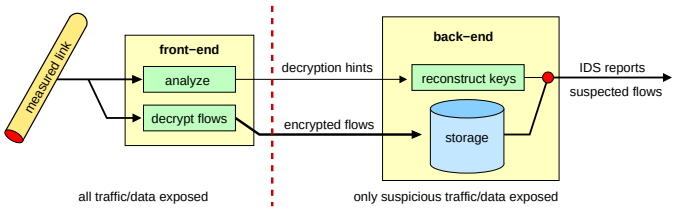


Fig. 1. Two-tier design for privacy-aware traffic monitoring.

B. Gradual Disclosure of Secret

Let us next briefly motivate the unconventional way to use the previously described secret sharing primitive.

1) Privacy-aware traffic monitoring: In [1] the authors consider the Shamir's secret sharing scheme as part of a network traffic anomaly detection system consisting of two active parts, as illustrated in Fig. 1. The so-called front-end encrypts the traffic flows in real time and, at the same time, it also detects potentially malicious events. Each flow has a unique flow identification number (ID), e.g., based on the traditional 5-tuple, and a corresponding encryption key (KEY). The so-called back-end stores the encrypted flows, and if an anomaly is suspected, it can "dig" further into the problem. The secret key escrow scheme is used to reveal (ID,KEY)-pairs gradually as suspicious events are observed. More specifically, upon observing an event corresponding to a suspected malicious activity, the front-end discloses a share of the corresponding secret to the back-end ("decryption hints" in Fig. 1). This way, the back-end will not be able to analyze any non-suspicious flows, which would violate users' privacy. That is, the raw data is exposed to the system in full only at the front-end, which task is to protect it in such a way that, given a strong enough reason, any relevant part of the information conveyed to the back-end, can be disclosed independently of the any other parts. Note that the secret key can stand for miscellaneous things including a malicious source IP address, decryption key for particular flows, etc. Moreover, there may not even be a need to deliver encrypted packet flows or flow data to the back-end, if, e.g., the source IP address is the only relevant information for the particular traffic monitoring application.

For a real time operation, it is essential that the computational burden in the front-end is kept minimal. The possible huge number of concurrent flows implies that keeping per flow state information is not feasible. To this end, the per flow keys can be computed from the flow IDs by an appropriately chosen pseudorandom function. Note also that the stateless operation facilitates multiple front-ends operating in distributed fashion.

2) Sensor networks: Similar counting mechanisms can be found useful in sensor networks consisting of large numbers of sensors [4]. Such networks can be used to monitor some activity such as movement. Assume that the task is to raise an alarm with the location information if at the given point more than m suspicious events have occurred. In this case, each activated sensor transmits information about the observed event by sending a share or shares of the secret eventually revealing the location of the corresponding nodes to a central monitoring point. The monitoring party can compute the corresponding location when he has received m shares from the sensor around the same location. In particular, there is no useful information available about the movement activity until the given threshold has been exceeded.

C. Reference Model

The reference model considered in this paper is as follows:

- *Sensor* reveals secret s once event e has occured m times.
- *Public* should not gain any information about the secret before the mth event.
- *Sensor* has limited processing capability and should be stateless (i.e., it cannot count itself).
- *Public* has less strict computational requirements.

Thus, the sensor knows s or can derive it from e. As events occur, sensor starts to leak or spray bits of information in such a manner that after the mth event the secret should be disclosable by the public. The elegant solution proposed in [1] achieves this by relying on the Shamir's scheme:

Basic Scheme:

1) *Upon detecting an event e, sensor sends a point $\{x_i, P(x_i)\}$, where the x_i is chosen in random, $0<x_i<p$.*
2) As soon as the public has received m (different) points, it can compute $P(x)$ and disclose the secret $s = P(0)$.

Note that in an ideal case, the sensor ensures that $x_i \neq x_j \ \forall \ i \neq j$. For $m \ll p$, choosing the x_i independently in uniform from $\{1, 2, \ldots, p-1\}$ already, in practice, guarantees this. Consequently, in this case the sensor disclosing the points does not need to keep any state information about the number of occurred events or the points disclosed, which clearly facilitates a stateless operation.

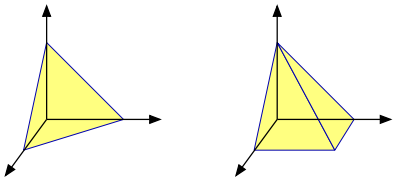
D. Several Threshold Type Criterion

The basic secret threshold counting scheme considered in [1] consists of a single counter, A, and the criteria for revealing a secret corresponding to the malicious event is of type

$$A \geq m.$$

A straightforward extension is to consider, e.g., two or more (independent) counters. Let B denote the second counter. Then the criteria can be,

1) $A + B \geq m$, which is achieved by letting A and B reveal shares from a same secret (sharing scheme).
2) $\min\{A, B\} \geq m$, obtained by using two independent shares that are combined in a higher layer shared secret (two levels). This can be generalized in a straightforward manner leading to a tree structure. Roughly speaking, this condition means "both A and B".
3) $\max\{A, B\} \geq m$, obtained by having two different shared secret configurations with equal secret, i.e., two polynomials $P_1(x)$ and $P_2(x)$ with $P_1(0) = P_2(0) = s$. This condition means "either A or B".



(a) (b)

Fig. 2. Hyperplane defined by a criteria (a), $S + A + B = m$, and a more complicated criteria (b) defined by, $S + \max\{A, B\} = m$.

Example 1: *Consider a case with one strong indicator, S, and two weak indicators A and B, each representing number of corresponding events, and two threshold criterion:*

1) $S + A + B \geq m$, *or*
2) $S + \max\{A, B\} \geq m$,

as illustrated in Fig. 2.

Note that randomness is a consequence of the requirement that the sensor is stateless. These types of conditions are straightforward to implement. For example, for the latter condition consider two polynomials, P_1 and P_2, defined in a shifted finite field consisting of the elements $\{-t, \ldots, p-1-t\}$, where t is an appropriately chosen constant. Define P_1 and P_2 so that they agree on the positive axis, and disagree on the negative axis. Each strong event S reveals a share from the positive axis, which is useful with respect to P_1 and P_2, and the secret can be disclosed as soon as $A + S > m$ or $B + S > m$. This type of combined criterion and multiple counters can be extremely useful and powerful in practice. In this paper we, however, do not pursue further in this direction, but instead focus on studying a single counter at its limit.

III. Analysis

Let us next consider certain cases where the basic scheme has difficulties. Firstly, consider a situation where the chosen threshold m is large compared to p. In this case, choosing the x_i uniformly in random eventually leads to clashes, i.e., it is likely that $x_i = x_j$ for some $i \neq j$. Secondly, publishing a large number of shares means a higher overhead, which may become an obstacle for certain applications. Moreover, large m means polynomials of degree $m-1$, i.e., computational complexity increases somewhat too. This type of obstacles can be tackled (at least) in two different and complementary ways:

i) One can use standard thinning approach by revealing one point after each event only with probability of q. This way, m can be kept small, $m \ll p$, and on average about m/q events are needed before the secret is disclosed.[1]
ii) Alternatively, one can accept a possibility that some $x_i = x_j$, resulting in some unnecessary overhead (which may or may not be acceptable depending on the situation).

Due to the random nature of the process, the unavoidable drawback in both cases is that the actual number of events required for the disclosure of the secret may in some cases be considerably larger than desired. Note, e.g., that i) allows counting up to ∞ as $q \to 0$. In general case, combining i) and ii), we have the following scheme:

Thinning Scheme:

1) *When a sensor detects an event, with probability of q it reveals one of the k available shares, $k < p-1$.*
2) *Probability of revealing share x_i is q_i,*

$$q_i = \begin{cases} q/k, & \text{when } 0 < i \leq k, \\ 1-q, & \text{when } i = 0. \quad \textit{(thinning)} \end{cases} \tag{2}$$

where "share" x_0 means that nothing is disclosed.
3) *Knowledge of m shares allows one to disclose the secret.*

Note that, as all points x_i are statistically identical, it is sufficient to consider a distribution as defined by (2).

Generalized version: In above, we have a single sensor guarding the secret and the number of shares k must be equal to m or greater, i.e., the sensor knows the secret and we could as well set $k = p-1$. However, in a more general case, the secret may be shared among several sensors working independently in distributed fashion. In such a case, a single sensor j may be assigned to work with $k_j < p-1$ shares in order to obtain a desired type of secret revealing process. For example, some sensors may provide a more reliable indication and thus should have a stronger weight in the process, which can be implemented simply by assigning them more shares. Similarly, $k_j < m$ means that a positive indication is required from more than one sensor, and moreover, a single sensor does not need to know the secret. As each sensors operates independently, the analysis of how the shares get disclosed remains essentially the same. Hence, for simplicity of presentation we limit ourselves to analyze a single disclosure process for the rest of the paper.

Example 2: *Pure thinning approach i) corresponds to $q_0 = 1-q$ and $q_i = q/(p-1), \forall\, i = 1, 2, \ldots, p-1$. In particular, for $m \ll p$ we have*

$$\mathrm{E}[M] \approx \frac{m}{q} \quad \textit{and} \quad \mathrm{V}[M] \approx \frac{m(1-q)}{q^2},$$

where a small q implies a large variance, which, depending on the particular application, may become an obstacle for using the thinning approach to extend the counting range.

It is easy to see that the corresponding secret disclosure process can be described as a Markov chain [5]. The current state is defined by the shares or points disclosed, and the transition probabilities are clearly independent of the previous states yielding a Markovian system. With this insight, the analysis of the disclosure process is rather straightforward. Let random variable M denote the number of steps occurring before the secret is disclosed, i.e.,

$$M = \text{number of steps before } m \text{ unique shares disclosed.}$$

By definition, $M \geq m$. The interesting quantities in this case are, e.g., $\mathrm{P}\{M = m\}$, i.e., the probability that disclosure occurs exactly after m steps, and the first two moments of

[1] Note that, in the limit $m = 1$, the Shamir's secret sharing scheme is actually replaced by explicitly revealing the secret with probability of q, and the disclosure occurs after $1/q$ events on average.

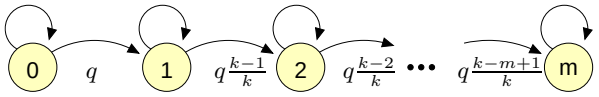


Fig. 3. Pure birth-process describing the disclosure of the secret according to thinning scheme.

M yielding the mean $\mathrm{E}[M]$ and the variance $\mathrm{V}[M]$, where the latter, in most cases, is minimized for obvious reasons.

Consider next the general case given by (2). The state space of the corresponding Markov chain can be reduced by state aggregation to a linear state space with states $0, \ldots, m$ as illustrated in Fig. 3 (state-dependent birth process). Let $\tilde{q}_i$ denote the birth-probability, i.e., the probability of moving from state i to state $i+1$, for which we have

$$\tilde{q}_i = q \cdot \frac{k-i}{k}.$$

Let T_i denote the number of events it takes to move from state i to state $i+1$. From Fig. 3 one immediately obtains,

$$M = T_0 + T_1 + \ldots + T_{m-1},$$

where $T_i \sim \mathrm{geometric}(q(k-i)/k)$, i.e., the T_i obey geometric distributions with parameters $q(k-i)/k$. In particular, the T_i are independent random variables. Consequently,

$$\mathrm{P}\{M=m\} = \frac{q^m k!}{(k-m)!\, k^m}. \tag{3}$$

This is illustrated in Fig. 4 (left) for $k = p_{16} - 1$ where p_{16} denotes the largest prime less than 2^{16}, $p_{16} = 2^{16} - 15$. The 16 bit long numbers are rather small with respect to the capabilities of the today's PC's, but can be argued for in many embedded solutions with limited computational capabilities.

Similarly, in Fig. 4 (right) we have depicted the maximum threshold values $m_{\max}$ for which the probability of clashes remains below 0.2%, 1% and 2% as a function of the number of available shares k. Observe that the maximal threshold values increase rather slowly as a function of k. Assuming the 1% clash probability is the design criteria, for $16/32$-bit arithmetic implementation ($k = p_{16} - 1$) the basic scheme allows counting up to a threshold $m \approx 35$, and relaxing the criteria a bit to 2% allows threshold up to $m \approx 50$.

It is also straightforward to see that the mean number of steps before the secret is disclosed, $\mathrm{E}[M]$, is given by

$$\mathrm{E}[M] = \frac{k}{q}(H_k - H_{k-m}), \tag{4}$$

where H_i denotes a partial sum of the harmonic serie, $H_i = \sum_{j=1}^{i} 1/j$, i.e., the ith harmonic number. Moreover, for the variance one obtains (sum of independent random variables),

$$\begin{aligned}\mathrm{V}[M] &= \frac{k}{q^2} \sum_{i=0}^{m-1} \frac{k(1-q)+iq}{(k-i)^2} \qquad (5)\\ &\geq \frac{1}{kq^2} \sum_{i=0}^{m-1} k(1-q)+iq = \frac{m(1-q)}{q^2} + \frac{m(m-1)}{2kq}.\end{aligned}$$

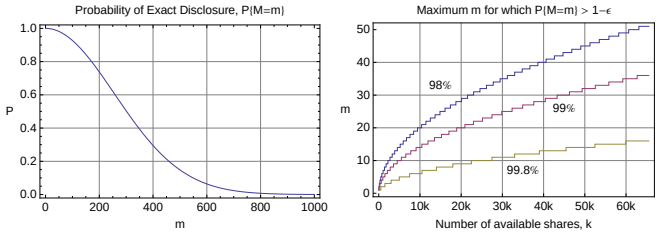


Fig. 4. Left: probability of exact disclosure, $\mathrm{P}\{M{=}m\}$, as a function of m for $q{=}1$ and $k{=}p_{16} - 1{=}2^{16} - 16$. ($p_{16}$ denotes the largest 16-bit prime). Right: maximal threshold values m for which $\mathrm{P}\{M = m\} > \tau$ for $q = 1$ and $\tau = 98\%, 99\%, 99.8\%$ as a function of available shares k.

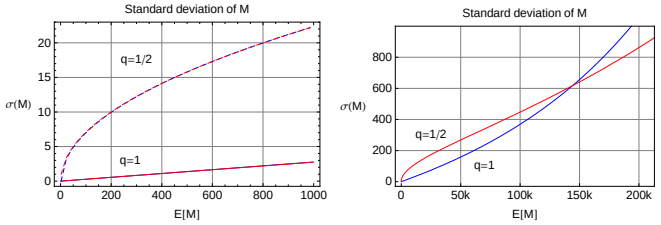


Fig. 5. Standard deviation of M as a function of $\mathrm{E}[M]$ for $k = p_{16} - 1$ and $q = 1, 1/2$. Note that as $\mathrm{E}[M] \gtrapprox 145000$, using thinning with $q = 1/2$ yields both a smaller variance and a smaller m (reporting overhead).

In general, using chain rules of expectation and variance for state-independent thinning by q yield

$$\mathrm{E}[M] = \frac{\mathrm{E}[M^*]}{q}, \text{ and } \mathrm{V}[M] = \frac{(1{-}q)\mathrm{E}[M^*]{+}\mathrm{V}[M^*]}{q^2}, \tag{6}$$

where M^* denotes the number of steps without thinning ($q{=}1$). From the above, it is clear that using thinning, $q{<}1$, leads to a significant increase in variance of M, i.e., $\mathrm{V}[M] \propto 1/q^2$ when q is small, and $\mathrm{V}[M] \propto 1/q$ for $q \approx 1$. From (5),

$$\mathrm{V}[M] \approx \frac{m(1-q)}{q^2} + \frac{m(m-1)}{2kq}, \quad \text{when } k \gg m, \tag{7}$$

in which case we have the following special cases:

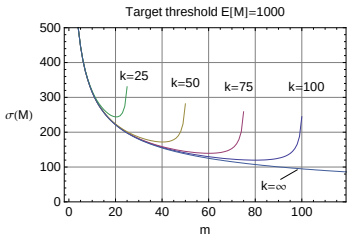
$$\mathrm{V}[M] \approx \frac{1-q}{q^2} \qquad \text{when } m = 1, \tag{8}$$

$$\mathrm{V}[M] \approx \frac{m(m-1)}{2k} \qquad \text{when } q = 1, \tag{9}$$

$$\mathrm{V}[M] \approx \frac{m(1-q)}{q^2} \qquad \text{when } k \gg m^2. \tag{10}$$

The last case corresponds to Example 2. In Fig. 5 we have depicted the standard deviation of M, $\sigma(M)$, as a function of $\mathrm{E}[M]$ when $k = p_{16} - 1$ and $q = 1/2, 1$. Without thinning $\sigma(M)$ is initially relatively small and at acceptable level for many applications. However, when the target threshold $\mathrm{E}[M]$ is higher than about 145000, the thinning approach becomes more reliable. This is due to the fact that without thinning the parameter m starts to approach k and hitting the last new shares may take some time.

Using (3) typically means that the scheme is required to count exactly to m before the disclosure. However, in many cases it is acceptable if the counting process involves a small degree of randomness. In this case, the mean $\mathrm{E}[M]$ is adjusted



k	m	$\sigma(M)$	q
25	20	244	0.038
50	40	172	0.079
75	60	139	0.119
100	80	120	0.159

Optimal values for $\mathrm{E}[M]$=1000.

Fig. 6. Standard deviation of M as a function of m when $\mathrm{E}[M] = 1000$.

accordingly and the variance $\mathrm{V}[M]$, corresponding to the randomness, is minimized by choosing the three parameters, m, k and q, appropriately. Thinning parameter q can be obtained from (4) as a function of $\mathrm{E}[M]$, m and k. Then the number of available shares, k, can be considered as given, leaving us with m which optimal value we are interested in. This is illustrated in Fig. 6, where we have chosen a target threshold of $\mathrm{E}[M] = 1000$ and $k = 25, 50, 75, 100$ shares. The curve with $k = \infty$ represents a lower bound when the thinning process is solely responsible for the randomness. Note that in each of the four cases with a finite pool of shares the optimal point does not deviate much from the lower bound, i.e., the thinning process is the dominating factor for the uncertainty.

In conclusion, the straightforward thinning extension of the basic scheme using random selection increases variance. For a finite number of shares, k, various combinations of threshold m and thinning probability q provide the same mean number of events $\mathrm{E}[M]$ to the disclosure. The optimal value for m for larger $\mathrm{E}[M]$ is generally always less than k, i.e., it makes sense to thin the process more than absolutely necessary. This also means that the degree of the polynomial guarding the secret is smaller than otherwise. In the next section we propose an alternative approach to mitigate the problem due to the random operation stateless operation by using rateless fountain coding type approach to minimize the variance.

IV. New Approach Based on Rateless Codes

To recap, we have a finite set of points $\{x_i, P(x_i)\}$, $i = 1, \ldots, k$, corresponding to a certain secret s, and k is small in comparison to the targeted $\mathrm{E}[M]$ (and $m \leq k$). Moreover, we want to minimize $\mathrm{V}[M]$, as a high variance corresponds to unreliable disclosure scheme, which in the worst case can take ages before revealing the secret. In this section we consider the possibility of using the fountain coding scheme (see, e.g., [6]–[9]) to decrease the variance $\mathrm{V}[M]$ considerably and thus improving the reliability of the counting scheme.

A. Fountain Coding

Assume that the original message consists of m blocks and the task is to transmit the message over an erasure channel (a channel where transmission is either successful, or nothing/error is received). In order to transmit the message reliably one can rely on forward error correction code such as Reed-Solomon. In particular, (n, m) Reed-Solomon codes can be used to generate $n > m$ symbols in a way that any subset of m symbols out of n would lead to disclosure of the original message. However, the Reed-Solomon codes are not practical for large n, m, while, in contrary, the so-called LT codes have extremely low computational complexity allowing efficient schemes for large values of m.

A so-called fountain coding principle states that any subset of $m + \epsilon$ received symbols should allow one to decode a message of length m. The name fountain coding stems from the fact that an encoder can generate practically infinite number of statistically identical packets (water drops), and the decoder can reconstruct the message (with high probability) as soon as it has received $m+\epsilon$ packets ("collecting water drops to a bucket"). The LT codes, proposed by Luby in [6], realize this principle and work as follows. First, the message to be transmitted is divided into m equally long blocks:[2]

$$X = \{x_1, x_2, \ldots x_m\}.$$

At each round, the encoder chooses d blocks from X in random, where the packet's degree d is drawn from a so-called degree-distribution. The chosen blocks are combined using exclusive-or operation and transmitted over to the recipient(s), i.e., letting the a_i, $i = 1, \ldots, d$, denote the uniformly in random chosen blocks, the packet to be sent is

$$x_{a_1} \oplus x_{a_2} \oplus \ldots \oplus x_{a_d},$$

where $\oplus$ (typically) denotes exclusive-or operator. The task of the receiver is to collect enough encoded packets in order to decode the original blocks. To be exact, the decoding process corresponds to solving a linear system of equations with m unknowns, which means that the decoding is possible as soon as m linearly independent packets have been received. Assuming each block is included independently with probability of $1/2$, then one can show that this *ideal decoder* yields

$$\mathrm{E}[\epsilon] < 2, \quad \text{and} \quad \mathrm{V}[\epsilon] < 3, \quad \forall\, m = 2, 3, \ldots \tag{11}$$

However, the proposed LT decoder relies on a considerably simpler approach and operates only with degree-1 packets (which correspond to the original blocks). Each time a new degree-1 block is discovered, it can be subtracted away from the other received packets including the discovered block. This way eventually more and more blocks are discovered, and the decoding process continues. The subtraction of known blocks is carried out also for each arriving packet. Thus, in order for decoding to start one needs at least one degree-1 packet. It is worth noting that the performance of the LT codes depends strongly and solely on the degree distribution, and, e.g., the optimal distribution for ideal decoder is not really practical. For large values of m, a so-called soliton distribution guarantees that on average only a small amount of additional packets (more than m) are needed to complete the decoding. For further details, we refer to [6]–[8].

B. Extending the Basic Scheme by LT codes

Next we describe a new hybrid approach which extends the basic scheme by combining it with a forward error

[2]The re-use of the same symbol m here is no coincidence.

correction scheme based on LT codes [6], [8]. There is, however, two fundamental differences between our scenario and the problem that LT codes are designed for. Firstly, the LT codes aim at delivering the whole message of m blocks, i.e., "$k = m = p - 1$". In contrast, in our case $m \leq k$, i.e., a successful event corresponds to a partially decoded message in standard LT coding scenario. In fact, any (n, m)-threshold secret sharing scheme can be seen as a special type of forward error correction (FEC), where efficiency is not such an important factor. Secondly, in secret sharing scheme it is of uttermost importance that no information about the secret leaks out until the grand moment. The gain of combining shares is best explained by a simple example:

Example 3: *Let $k = 2$, i.e., sensor has two shares, $P(x_1)$ and $P(x_2)$, where $P(x) = a_1 x + a_0 \mod p$. The basic scheme discloses either $P(x_1)$ or $P(x_2)$ after each event, and an example realization could be $\{P(x_1),\ P(x_1),\ P(x_2),\ \ldots\}$, leading to disclosure of the secret in three steps. The disclosure probability after two steps is $1/2$. In an alternatively hybrid approach, the sensor discloses $P(x_1)$, $P(x_2)$ or $P(x_1) \oplus P(x_2)$, and, e.g., a realization $\{P(x_1),\ P(x_1) \oplus P(x_2),\ \ldots\}$ already reveals both $P(x_1)$ and $P(x_2)$, and consequently also the secret. Assuming that after each event one of the three shares is chosen uniformly in random, it follows that the probability of disclosing the secret after two rounds is $2/3$ instead of $1/2$ obtained with the basic scheme.*

The important property with the Shamir's scheme is that knowledge of $P(x_1)$ yields no further information about $P(x_2)$, i.e., $P(x_1)$ and $P(x_2)$ are independent and the conditional distribution $\mathrm{P}\{P(x_2){=}i \mid P(x_1){=}j\}$ remains uniform. However, in our case, the exclusive-or operation can be extremely harmful. In Example 3 with $x_1{=}1$, $x_2{=}2$ and $p{=}5$, revealing a single share of $P(x_1) \oplus P(x_2) = 7$ implies that $P(x){=}4x$ or $P(x){=}x + 2$ leaving only two possible values left for the secret, i.e., $a_0{=}0$ or $a_0{=}2$. This deficiency can be overcome by using addition modulo p instead of the exclusive-or. In general, a modulo p sum of k independent random variables $X_i \in \{0, \ldots, p-1\}$ is uniformly distributed and independent of any partial sum of the X_i. Consequently, the hybrid scheme possesses the same property of not "leaking out" any information about the secret before the mth linearly independent share has been revealed. Thus, we have:

Hybrid Scheme:

1) *Upon detecting an event, the sensor draws a random degree d from the degree distribution.*
2) *Sensor chooses d random shares $\{x_{i_1}, \ldots, x_{i_d}\}$, and reveals $\sum_j P(x_{i_j}) \mod p$.*
3) *Public first decodes the individual shares $P(x_i)$ by subtracting known points from the combined.*
4) *When m or more points of the polynomial $P(x)$ are known, the public can disclose the secret $s = P(0)$.*

It is more probable that the combined shares can be decoded when $m \approx k$, or even $m = k$. Moreover, when a secret corresponds, e.g., to a potential malicious traffic source, it is clear that these events do not (normally) occur frequently. Therefore, it may even be desirable that an additional work in form of solving a system of linear equations needs to be accomplished before the secret is revealed (cf. key strengthening), and the corresponding traffic flow can be analyzed. In particular, solving such an equation for a moderate number of unknowns is feasible:

Example 4: *Assume $k{=}m{=}25, 50, 75, 100$, target $\mathrm{E}[M] = 1000$ and an ideal decoder. Substituting* (11) *into* (6) *gives $\sigma(M) \approx 201, 139, 112, 95$, respectively. Hence, a clear improvement is achieved (cf. Fig. 6), while there is still no way to avoid the inaccuracy due to the thinning procedure.*

To further elaborate the hybrid approach, let us next describe another similar scheme ("a toy example"), where shares are paired and occasionally a combined pair is also disclosed.

Example 5: *Consider a simple pairing scheme where the original number of shares is an even number, $k = 2n$. Then one can pair the $2n$ shares in some manner,*

$$(a_1, b_1),\ (a_2, b_2),\ \ldots (a_n, b_n).$$

The basic scheme in this case works by first choosing the pair i in random, and then either a_i or b_i. Two stage approach is functionally equivalent to the original scheme. In an alternative pairing scheme, we extend each pair to a triple,

$$(a_1, b_1, a_1 \oplus b_1),\ (a_2, b_2, a_2 \oplus b_2),\ \ldots (a_n, b_n, a_n \oplus b_n).$$

Otherwise the procedure remains the same, first choose triple i in uniform, and then one of the three choices again in uniform.

For case $n = 1$ and $m = 2$, the pairing scheme improves the performance (see Example 3). It turns out that both schemes can be analyzed and expressions for the exact disclosure probability, $\mathrm{P}\{M = m\}$, can be derived. For the basic case, (3) yields

$$\mathrm{P}\{M = m\} = \frac{(2n)!}{(2n-m)!\,(2n)^m} = \binom{2n}{m} \frac{m!}{(2n)^m}, \quad (12)$$

and for the pairing scheme we have (see Appendix)

$$\mathrm{P}\{M = m\} = \sum_{i=i_{\min}}^{i_{\max}} \binom{n}{i} \binom{n-i}{m-2i} \frac{m!}{n^m} \cdot \frac{2^{m-2i}}{3^{m-i}}, \quad (13)$$

where $i_{\min} = \max\{0, m{-}n\}$ and $i_{\max} = \lfloor m/2 \rfloor$. With respect to this criteria, the pairing scheme is better than the basic scheme when the counting threshold m is about 70% or more of the k, which is illustrated in Fig. 7. The hybrid pairing scheme is better in the higher blue triangle, and the basic scheme in the lower yellow triangle.

The first few cases for $\mathrm{P}\{M = m\}$ are also given in Table I. We note that this extreme case, where each share is required for a successful disclosure, is the most favorable for the hybrid approaches, and consequently, even this simple pairing scheme offers huge improvement over the basic scheme as the number of shares m increases. Moreover, with these two schemes the probability $\mathrm{P}\{M = m\}$ is very small already for $m \geq 20$ making it impractical (and also, hard to validate by numerical simulations). Consequently, we should have $m < k$ for all practical purposes with these two schemes.

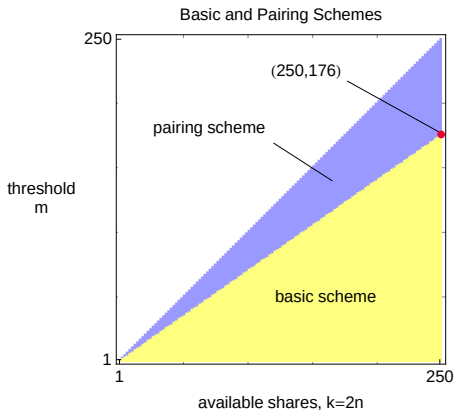


Fig. 7. Optimal choice between the basic scheme (lower yellow area) and hybrid pairing scheme (upper blue area) for $1 \leq m \leq k \leq 250$. It appears that as soon as m is about 70% or more of k, then the pairing scheme becomes superior. Before that the risk of individual clashes is smaller than the risk of choosing isolated combined shares.

V. Conclusions

In this paper we have analyzed a new interesting application of the Shamir's secret sharing algorithm to escrow a secret key as a result of a *counting process* [1]. The basic scheme is defined by threshold m (degree of the secret sharing polynomial is $m-1$), prime number p defining the finite field, and the number of available shares k, where $m \leq k < p$. For $m \ll k$, the basic scheme works flawlessly and allows counting up to m in a way that the mean number of steps $\mathrm{E}[M] \approx m$. In contrast, our focus has been in cases where one, for some reason, wants to increase the counting target $\mathrm{E}[M]$ beyond the limits of the basic scheme. In this case the randomness comes into play. Thinning the events by a constant probability of q still facilitates a stateless operation, while allowing one to count to any number (on average) at the same time. Randomness, however, introduces also variance in M which translates to an inaccuracy in the counting process. We have analyzed this process and give expressions that allow one to choose the optimal values for the parameters that minimize the variance.

Moreover, we have also proposed a hybrid scheme in order to mitigate the reliability problem when the threshold m is close to the number of shares k. The hybrid scheme combines the ideas of the LT codes with the original scheme, and can provide a considerable improvement in the variance when $m \approx k$. The complexity increases only minimally, and in particular, the nature of stateless operation is not changed, which is the fundamental requirement, e.g., for the key escrow in a traffic monitoring system as proposed in [1]. With respect to the design of the degree distribution for the LT codes, the objective function in this context can be seen as the generalization where the decoding process is considered to be successful when $m \leq n$ blocks have been recovered (instead of the full message, $m = n$). The design of such a degree distribution is also an interesting topic for future research.

TABLE I
Comparison of the basic scheme to the pairing scheme. The gain from the pairing increases considerably as m increases.

length $m = k$	Success probability $\mathrm{P}\{M = m\}$ basic	pairing	improvement from pairing
2	$1/2$	$2/3$	33%
4	$3/32$	$1/6$	78%
6	$5/324$	$80/2187$	137%
8	0.0024	0.0076	216%

Acknowledgements

The Telecommunications Research Center Vienna (ftw.) is supported by the Austrian Government and the City of Vienna within the competence center program COMET. This work was done within the scope of the EU FP7 project PRISM (grant no. 215350), and partially supported by the strategic project N0 at ftw.

References

[1] G. Bianchi, S. Teofili, and M. Pomposini, "New directions in privacy-preserving anomaly detection for network traffic," in *1st ACM Workshop on Network Data Anonymization (NDA 2008)*, Oct. 2008.
[2] B. Schneier, *Applied Cryptography; Protocols, Algorithms and Source Code in C*, 2nd ed. John Wiley & Sons, 1996.
[3] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, 2002.
[5] S. M. Ross, *Introduction to Probability Models*, Academic Press, 2000.
[6] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Nov. 2002, pp. 271–280.
[7] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2004.
[8] D. MacKay, "Fountain codes," *IEE Proc. of Communications*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.
[9] E. Hyytiä, T. Tirronen, and J. Virtamo, "Optimal degree distribution for LT codes with small message length," in *IEEE Infocom mini-symposium*, Anchorage, Alaska, USA, May 2007, pp. 2576–2580.

Appendix

Here we analyze the pairing scheme of Example 5, and derive (13) giving the probability that the secret can be disclosed after exactly m steps. Let ℓ denote the number of choices in each "bin" (two or three), where $\ell = 2$ is equivalent to the basic scheme with no combined share, and $\ell = 3$ corresponds to the pairing scheme with three types of shares. For the basic scheme, $\ell = 2$, we already have (12) as a corollary of (12). For the pairing scheme with $\ell = 3$, the situation is somewhat more complicated.

Let us next consider the general case with $\ell = 2$ or $\ell = 3$, even though (12) already holds for $\ell = 2$. There is a clear analogy to placing m balls into n bins and assigning each of them with one of the ℓ colors. We will use this terminology, and by colors red, blue and green we refer to a_i, b_i and $a_i \oplus b_i$, respectively. In case $\ell = 2$, we have only red and blue balls. Let random variable I denote the number of bins with two balls. Note that if any bin has more than two balls, then there is necessarily a clash and $M > m$. Thus, $I = i$ means that i bins have two balls and $m - 2i$ bins have one ball. Consequently,

the parameter i can have values from $i_{\min}$ to $i_{\max}$,

$$i_{\min} = \max\{0, m-n\}, \quad \text{and} \quad i_{\max} = \lfloor m/2 \rfloor.$$

Conditioning on the event that $I = i$, gives

$$\mathrm{P}\{M = m\} = \sum_{i=i_{\min}}^{i_{\max}} \mathrm{P}\{I = i\} \cdot \mathrm{P}\{M = m \mid I = i\}.$$

The latter probability for $\ell = 2$ is simply

$$\mathrm{P}\{M = m \mid I = i\} = \left(\frac{1}{2}\right)^i.$$

For $\ell = 3$, the bins with a single ball can also cause a problem if the ball is painted with green color (i.e., a combined share). Hence,

$$\mathrm{P}\{M = m \mid I = i\} = \left(\frac{2}{3}\right)^i \cdot \left(\frac{2}{3}\right)^{m-2i} = \left(\frac{2}{3}\right)^{m-i}$$

The former probability, $\mathrm{P}\{I = i\}$, can be derived by combinatorial means. First note that the bins can be chosen in

$$\binom{n}{i}\binom{n-i}{m-2i},$$

different ways. Then, the m balls can be assigned in

$$\left[\ell^{m-2i}(m-2i)!\right]\left[\ell^{2i}\frac{(2i)!}{2^i}\right]\binom{m}{2i},$$

different ways to these given bins, where the first part corresponds to assigning single ball to $m-2i$ bins, the middle part to assigning two balls to i bins, and the last binomial coefficient to the number of ways one can split m to $2i$ and $m-2i$ groups. The total number of possible combinations is $(\ell \cdot n)^m$. Combining these, after some manipulations, yields

$$\mathrm{P}\{I = i\} = \binom{n}{i}\binom{n-i}{m-2i}\frac{m!}{n^m}\left(\frac{1}{2}\right)^i.$$

Hence, the final formulæ are

$$\mathrm{P}\{M = m\} = \begin{cases} \displaystyle\sum_{i=i_{\min}}^{i_{\max}} \binom{n}{i}\binom{n-i}{m-2i}\frac{m!}{n^m} \cdot \frac{1}{4^i}, & \text{if } \ell = 2, \\ \displaystyle\sum_{i=i_{\min}}^{i_{\max}} \binom{n}{i}\binom{n-i}{m-2i}\frac{m!}{n^m} \cdot \frac{2^{m-2i}}{3^{m-i}}, & \text{if } \ell = 3. \end{cases}$$

where the case $\ell = 2$ is equal to (12), and the case $\ell = 3$ corresponds to (13).