Framework for Managing Threshold Cryptography in Distributed Network Environments

Dr. Ghassan Chaddoud[*] Prof. Keith Martin[**]

Abstract

An increasingly common requirement in distributed network environments is the need to distribute security mechanisms across several network components. This includes both cryptographic key distribution and cryptographic computation. Most proposed security mechanisms are based on threshold cryptography, which allows a cryptographic computation to be shared amongst network components in such a way that a threshold of active components are required for the security operation to be successfully enabled. Although there are many different proposed techniques available, we feel that the practical issues that determine both what kind of technique is selected for implementation and how it is implemented are often glossed over. In this paper we thus establish a new framework for network security architects to apply when considering adoption of such mechanisms. This framework identifies the critical design decisions that need to be taken into account and is intended to aid both design and implementation. As part of this framework we propose a taxonomy of management models and application environments. We also demonstrate the utility of the framework by applying it to a VPN environment.

Keywords: threshold cryptography, security architecture, distributed environments, management odel.

[*] Atomic Energy Commission of Syria-P.O.Box 6091, Damascus
[**] Royal Holloway, University of London-Egham, Surrey TW20 0EX, UK

1. INTRODUCTION

In recent years the increasing focus on distributed network environments has resulted in many new classes of application where traditional approaches to security, through the direct application of cryptographic primitives, cannot be applied in the usual way. In such environments a common requirement is that instead of one single network entity conducting a cryptographic operation (at any of the key distribution, transmission, or reception stages), groups of entities are required to jointly conduct an operation. For example, the creation of a network master key might require several high level network components to jointly contribute keying material, or the signing of a message sent by a group might require cooperation of a minimum number of signing entities in order to authorize the signature.

There are many different types of security mechanism that have been designed for application in such environments and these have motivated the development of the theory of distributed cryptography, often referred to as threshold cryptography [1]. Such mechanisms are typically based on threshold schemes (sometime also referred to as shared control systems, secret sharing schemes, or concurrency schemes) [2], [3]. A threshold scheme is essentially a technique for protecting a high value piece of information (or secret) by distributing knowledge related to that information across a number of network entities (often referred to as participants or shareholders). It is assumed that the secret is of such importance that no entity alone can be trusted to know it. The basic idea is that the entities must co-operate by combining privately held related information (shares of the secret) in order for the secret to be reconstructed or activated. These shares must be created and distributed to shareholders in a secure way. While threshold schemes can be applied directly to applications such as key distribution, it is common to integrate them into other cryptographic operations to establish mechanisms for operations such as threshold encryption [4], [5], [6] group signatures [7], [8], [9], [10] shared key generation [11] etc.

It is not uncommon for designers of network architectures or key management systems to identify areas where the application of threshold cryptography is appropriate (often as an extension to a proposed system), but to merely flag its potential and omit practical details. We feel that this is a mistake because putting threshold cryptography into practice schemes requires careful consideration of many important issues. These include both security architecture issues (roles, communication channels, etc) and scheme management issues (distribution of share information, share combination procedures adding and removing of shareholders, etc).

The application environment plays a crucial role in determining the security infrastructure and identifying the most appropriate way for scheme management operations to be carried out. In this context, we propose in this paper a framework for implementing threshold cryptography within any network environment. This framework identifies the issues that need to be addressed when designing an appropriate solution based on threshold cryptography. It is important to recognize that our framework identifies the critical design decisions that need to be taken into account and is intended to aid both design and implementation. However due to the large number of different mechanisms available in the literature, this framework only acts as a guide to mechanism selection. A prescriptive identification of the most appropriate security mechanism for each application is beyond the scope of this paper

The rest of this paper is organized as follows. Section 2 briefly introduces threshold schemes and some simple management operations. Section 3 presents a taxonomy of scheme management models. In Section 4 we outline a taxonomy of application environments. In Section 5 we present a list of scheme management issues that need to be addressed. Finally in Section 6 we apply the framework to demonstrate its utility in designing an architecture for a VPN environment supporting mobility.

2. BASIC THRESHOLD CRYPTOGRAPHY OPERATIONS

In this section we identify the basic operations of a typical threshold cryptography scheme. Before doing so we need to introduce the three abstract entities involved in most threshold cryptography schemes:

- The *dealer* is typically responsible for the creation and distribution of the shares (and may be involved in other operations such as shares refreshing).
- The *combiner* is responsible either for combining shares to recover the shared secret or engaging in computation that applies the shared secret.
- The *shareholders* are responsible for share maintenance, verification, and use.

Note, however, that the precise separation and responsibilities of each entity is application dependent. In some environments applications have to rely on shareholders to replace the dealer, combiner, or both. This might be for architectural reasons (no such third party entity exists or is available), or security reasons (no single entity can be trusted to perform the role).

A. Initialisation

Most (but not all) schemes assume the existence of a trusted dealer to conduct the initialisation process. For purely illustrative purposes we demonstrate the initialisation process for the well-studied *Shamir threshold scheme* [3]. This scheme forms the basis of many more complex threshold cryptography primitives.

Shamir's k-out-of-n or (k, n)-*threshold scheme* is designed to split a secret s into n shares in such a way that any k of these shares are sufficient to recompute the secret. The dealer constructs a secret polynomial $f(x)$ of degree at most $k - 1$ over Z_p (where p is a prime and bigger than n) such that $f(0) = s$. The dealer associates each of the n shareholders P_i ($1 \leq i \leq n$) with a public value x_i and assigns the secret value $y_i = f(x_i)$ (mod p) to P_i as their share. The dealer sends share y_i to participant P_i over a secure channel.

B. Secret reconstruction

In any (k, n)-threshold cryptography scheme, at least k shareholders must be involved in order to conduct the cryptographic process. This joint activity typically involves one of two processes.

In the first method, shareholders pool or combine their shares (via the combiner) in order to reconstruct the secret. The combiner then typically applies the secret to the desired cryptographic operation. For example, in the Shamir threshold scheme, if a subset B of k participants want to reconstruct the secret, they submit their shares to the combiner, who uses Lagrange interpolation to compute:

$$K = f(0) = \sum\nolimits_{j,P_j \in B} y_j \text{, where}$$

$$b_j = \prod_{i,P_i \in B, i \neq j} \frac{x_j}{x_j - x_i}$$

On the other hand, any combination of $k - 1$ or fewer shares do not reveal any information about the secret.

In the second method, which is typical of threshold encryption/decryption processes, the secret is never explicitly reconstructed, but is applied in a distributed way. In such cases a threshold of shareholders apply their shares independently to produce partial results. These are then sent to the combiner who conducts a joint computation and outputs a final result.

Figure 1 depicts both methods being applied to establish threshold variants of a DSS signature. In the first method (a) the combiner reconstructs a signature key K from the shares, which is used to sign a message, M. In the second method (b) each shareholder submits a partial signature to the combiner who combines them to form a threshold signature on the message.

In order to clarify this difference, we provide a few more details: Let p and q be two large primes such that q divides $p - 1$ and g generates the subgroup Z_q of Z_p of order q. Let h be a hash function whose range is {1, .., q−1}. The values p, q, g are public parameters, the private signature key is chosen to be x Z_q and the corresponding public verification key is $y = g^x$.

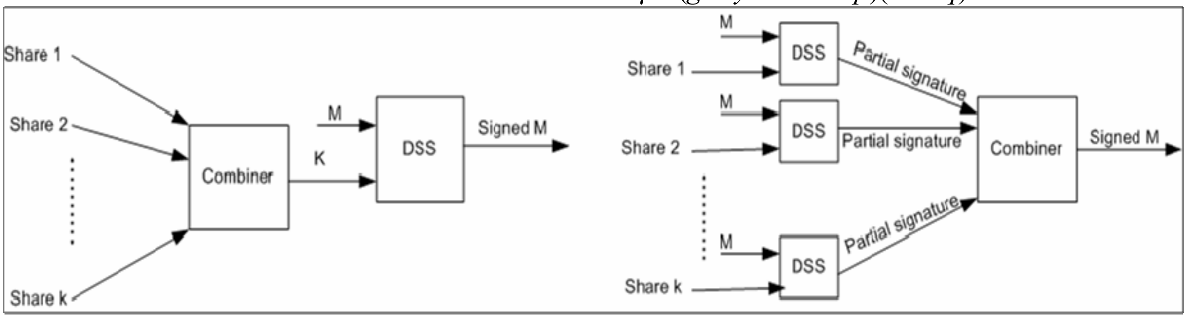
1. Method 1

Share the private signature key x amongst the n shareholders using (for example) the Shamir threshold scheme. To create a threshold DSS signature on the hashed message $m = h(M)$:

1. At least k shareholders present their shares y_i to the combiner, who reconstructs x.
2. The combiner creates the signature (γ, δ), where

$\gamma = g^r$ mod q (r is a random number in Z_q)
$\delta = \gamma.x + m.r \ mod\ q$.

The verifier notes that $ver\ (m, \gamma, \delta)$ = true ⇔ $\gamma = (g^{\delta/m} y^{-\gamma/m}\ mod\ p)(mod\ q)$.



(a) method 1

(b) method 2

Figure 1. Threshold DSS signature.

2. Method 2

Share the private signature key x amongst the n shareholders using (for example) the Shamir threshold scheme. The following scheme is based on one proposed in [9]. To create a threshold DSS signature on the hashed message $m = h(M)$:

1. At least k shareholders run a distributed key generation algorithm (details omitted). The result is that each of the k shareholders P_i has a share r_i of r, and a public value $\gamma = g^r$ (r itself will not be revealed).

2. Each shareholder computes
$\delta_i = \gamma.x_i + m.r_i \ mod\ q$,
and sends it secretly to the combiner.

3. The combiner computes γ by interpolation on the set of δ_i to produce the final signature (δ, γ).

C. Verification

At heart, a threshold cryptography scheme simply involves initialisation and secret reconstruction. However most practical schemes require a level of robustness against aliciousness of scheme participants. There are various levels of protection that can be built into threshold schemes and all come under the general heading of *verifiable secret sharing* [12], [13], [14].

For illustration only, we demonstrate an extension to the Shamir threshold scheme that offers limited robustness in that it allows shareholders to verify that their shares are consistent (*i.e.*, that any subset of k shares can determine the same unique key). This provides protection against distribution errors or malpractice by the dealer. To do this, prior to share distribution, the dealer broadcasts *witnesses*. After constructing the secret polynomial
$f(x) = a_0 + a_1x + ... + a_{k-1}x^{k-1}$,
the dealer makes publicly available the values g^{a0}, g^{a1}, ..., g^{ak-1}. When shareholder P_i receives a share y_i, they verify that

$g^{s_i} = g^{a_0}.(g^{a_1})^{id_i}...(g^{a_{k-1}})^{id_i^{k-1}}$,where id_i is an identifier of shareholder i.

If this equation does not hold then the shareholder has detected an error. For other examples of more robust verifiable threshold schemes, see for example [12].

D. Refreshment

To increase robustness against certain types of attack (such as a mobile adversary who tries to sequentially compromise a set of shareholders) it is good practice to periodically refresh shares. Schemes for refreshing shares are sometimes referred to as *proactive secret sharing schemes* [15], [16].

One method for refreshing shares in a Shamir threshold scheme is based on the observation that the Shamir threshold scheme is *linear*: if (y_1, . . ., y_n) is a set of shares for secret s_1 and (z_1, . . . , z_n) is a set of shares for secret s_2 then (y_1+ z_1, . . . , y_n+ z_n) is a set of shares for secret $s_1 + s_2$. Thus if a (k, n)-threshold scheme is currently in place with secret s based on polynomial f_{old}, the dealer can refresh shares but keep the same secret by broadcasting a new polynomial $f_{refresh}$ of degree at most $k - 1$ with

$f_{refresh}(0) = 0$. If each shareholder computes $f_{old}(P_i) + f_{refresh}(P_i)$ then they will have a new share in a (k, n)-threshold scheme based on $f_{new} = f_{old} + f_{refresh}$, whose secret is still s. The original shares can then be deleted.

E. Update operations

In dynamic environments the nature and relationship between shareholders may change at any time. These are either reflected in:

1. Changes in n: it might be desirable that new entities be added (enrolled) or evicted (disenrolled) from the set of current shareholders.
2. Changes in k: a change in the organizational policy might mandate an increase or decrease in the number of shareholders required to reconstruct the secret.

Note that these events might happen for a number of reasons: an entity may become corrupted, an entity may come online or go offline, etc. All of these events however lead to some change in the defining parameters of a (k, n)-threshold scheme.

With an online dealer and secure channels to all shareholders then all these operations are relatively straightforward (albeit inefficient) and roughly correspond to a rerun of the initialization operation. However in many application environments this will not be the case and there are many different proposed protocols for conducting update operations in more restrictive environments, for example [17]–[22]. Determining the operational environment in which update operations can be conducted is a major design decision when implementing threshold cryptography of any kind, and we consider the different options in Section 3.

3. SCHEME MANAGEMENT MODELS

In this section we propose a taxonomy of scheme management models for a threshold cryptography scheme. It is very important to identify which model is most appropriate for an application environment before selecting security mechanisms. Our taxonomy, which is summarised in Table 1, is based on the different environments in place during the *initialization phase* (during the initialisation operation) and the *running phase* (after the scheme has been set up, when refreshment and update operations take place) of a threshold cryptography scheme. Clearly there are many potential applications where these environments may be quite different (see Section 4). The taxonomy is defined by active entities and communications channels.

1. **Active entities**: We assume that shareholders and combiners are active throughout the lifetime of the scheme. However we do not assume that there exists a (potentially expensive) trusted dealer entity throughout the lifetime of the scheme. Four cases are distinguished:

a) The dealer remains active throughout the scheme lifetime.

b) The dealer is active during the initialisation phase and remains alive during the running phase, but with limited capabilities.

c) The dealer is active during the initialisation phase but not during the running phase.

d) There is no dealer during the initialisation or the running phase (in which case all scheme management is collectively done by shareholders).

2. **Communication channels**: We consider the communication channels that are in place between dealer and shareholders and between shareholders. We do not consider communication channels between shareholders and combiner as a distinguisher since this is almost always necessarily a secure channel. Three types of channel are distinguished:

a) Secure channel (offering authentication and confidentiality).

b) Public channel (offering authentication only).

c) No channel exists.

We do not distinguish between the nature of the channel for the purposes of this taxonomy (but see Section 5 for further discussion).

Based on the above distinguishers, in Table 1 we identify twelve meaningful scheme management models (these are organized into four blocks of rows corresponding to the four active entity classes defined above). These models are not the only possible configurations but they are the most plausible. The notation used in Table 1 is as follows:

- *Entities*: D: dealer, U: shareholders.
- *Links*: D - U: link between dealer and shareholder, U - U: link between shareholders.
- *Activity*: A: active, X: not active.
- *Channel*: S: secure, P: public, -: no channel.

From Table 1 we see that models 1 - 9 are *autocratic* in that they are all based on having an active dealer with secure links to all shareholders during the initialisation phase (hence there is no obvious need for shareholder to shareholder communication during initialisation). However these models all differ in the subsequent running phase environment. For example, in model 4 the dealer remains active but can only use a public channel to communicate with shareholders, whereas in model 7 the dealer is no longer active but shareholders have access to secure channels in order to conduct running phase operations. On the other hand, no dealer exists in models 10 - 12, which all rely on secure shareholder to shareholder links during the initialisation phase. We refer to these models as *democratic*.

Table 1. Scheme management models.

Model	Initialization phase			Running phase		
	D	D-U	U-U	D	D-U	U-U
1	A	S	-	A	S	S
2				A	S	P
3				A	S	-
4	A	S	-	A	P	S
5				A	P	P
6				A	P	-
7	A	S	-	X	-	S
8				X	-	P
9				X	-	-
10	X	-	S	X	-	S
11				X	-	P
12				X	-	-

Notes

1. During the running phase, certain update operations such as adding new shareholders clearly require the use of some secure channels. Hence models 5, 6, 8 and 9 cannot directly support such operations. It is possible however for a hybrid model to apply in such cases. For example, we might have a model 7 environment applying to existing shareholders but retain a dealer entity solely for initialisation of new shareholders.
2. Models 9 and 12 do not offer any possibility of running phase operations. If there is a need to refresh shares, for example, then any scheme operating in these models requires re-initialisation.
3. The capability within a model of conducting running phase operations does not by default imply that these can always be performed. For example, some schemes offer only limited capability for conducting running phase operations and require re-initialisation after a specified period (we discuss this issue in more detail in Section 5).

4. APPLICATION ENVIRONMENTS

In this section we propose a classification of plausible application environments for threshold cryptography. We also indicate which scheme management models are likely to be most suitable in each case. We identify five application classes:

1. **Class A**: This class is characterized by relatively small parameters (k and n) and a controlled implementation environment where the communication infrastructure is typically provided by secure physical links. Examples of applications include distributing components of a security access code to a bank vault or missile launch process, or component-wise generation of top-level cryptographic master keys for a multi-level banking key management hierarchy. This is the class of applications for which most early designers of threshold schemes envisaged their use. Such applications are most likely to conform to management model 9, however model 3 is also possible.
2. **Class B**: This class is characterized by a controlled implementation environment but differs

from Class A by not relying on physical links and thus fewer requirements for reduced parameter sizes. Such applications are typically running in a dedicated secure environment (such a dynamic virtual private network) and maintain a dealer entity throughout operation. Example applications include group signature schemes or approval of transactions in a banking network. While any of management models 1 to 6 are possible, models 3 and 6 would appear most plausible as maintenance of a dealer relaxes the need for inter-shareholder communication.

3. **Class C**: This class is characterized by a relatively controlled environment where certain

4. operations can be delegated to normal users. This includes limited-area communication groups such as virtual audio and video conferences, which are operating within open public environments such as the Internet and have a predetermined membership. Example applications include establishing cryptographic keying material to secure communications. As such groups are structured and controlled it is most likely again that management models 1 to 6 could all apply. However in contrast to Class B, in this case models 1, 2, 4 and 5 are more plausible.

5. **Class D**: This class is characterized by a controlled environment during the initialization phase, followed by an uncontrolled environment during the running phase. This includes wide-area communication groups where the dealer is aware of initial shareholders but, for reasons of scalability, hands over scheme management to shareholders during the running phase. An example application includes a distributed certificate authority facility within an ad-hoc networking environment. For this class management model 7 is most plausible, with models 8 and 9 are also possible.

6. **Class E**: This class is characterized by the lack of a controlled implementation environment. In this case all operations, including initialisation, must be distributed amongst shareholder entities. Applications in this class are typically operating in hostile dynamic environments, such as ad-hoc networks, where shareholders are likely to be fairly short-lived and mobile. An example application is group key generation by components for a cluster of nodes in an ad-hoc network. Such applications necessarily conform to management models 10, 11 or 12, with model 10 being by far the most plausible (as without any secure links any threshold application will be highly constrained).

The mapping of application classes to management models is summarised in Table 2, with possible models listed and most plausible models indicated in bold.

Table 2. Mapping between application classes and management models.

Applications	Model
Class A	3,**9**
Class B	1, 2, **3**, 4, 5, **6**
Class C	**1**, **2**, 3, **4**, **5**, 6
Class D	**7**, 8, 9
Class E	**10**, 11, 12

5. SCHEME MANAGEMENT ISSUES

In this section we identify a full range of issues that need to be addressed before selection of any threshold cryptographic mechanism. We raise each set of issues as a list of informed questions that need to be considered before making a design decision. The first two sets of issues are related to the application environment and hence are important in determining the scheme management model. The third and fourth sets of issues are designed to help identify the required operations. Finally, the last set of issues relate to the ability of involved entities to conduct operations.

A. Roles and responsibility

The first set of issues concerns the roles and responsibilities of entities in the scheme and provides important information towards selection of a scheme management model. Recall that there are three roles involved (dealer, combiner and shareholders), all of whom must exist in concept, but not necessarily as dedicated independent entities.

1. Is there a dedicated trusted dealer entity during the initialization phase?
- If not, then how is the dealer entity represented during the initialization phase?
2. Which entity is responsible for setting security policy (for example setting or changing the threshold level *k*) during the lifetime of the scheme?
3. Is there a dedicated trusted dealer entity during the running phase?

If not, then how is the dealer entity represented during the running phase?

4. During the running phase, which management operations are required and which entities are responsible for conducting them?
5. How is the combiner entity represented (is it the dealer, a nominated shareholder, a group of shareholders, or a dedicated security system)?
6. What trust assumptions apply to and between the different entities (for example, are entities trusted to securely destroy secret information)?

B. Networking environment

The next set of issues concern the environment within which the scheme will be implemented.

1. What communication links is it practical to establish between scheme entities during the initialization phase?
2. What communication links is it practical to establish between scheme entities during the running phase?
3. What security services are required on these links?
4. What security architecture will be available for implementing the scheme?
5. Where tradeoffs between security and performance are possible, what is the cost of establishing secure links and what is the impact on application performance?

C. Update operations

An important set of questions relates to the nature of any update operations that might be required of the implemented scheme.

1. To what extent is it possible during the initialization phase to predict the nature of any update operations that might take place during the running phase? This nature could be of the form:
   - Whether updates are likely to occur at all;
   - The number of updates that are likely to occur;
   - The classes of update that are likely to occur (e.g. disenrollments in general);
   - Precise updates that are likely to occur (e.g. disenrolling a particular participant).
2. During the running phase, what different update operations are required?
3. During the running phase, which entities are responsible for conducting update operations?
4. Is it acceptable to limit the number of update operations that can take place during the running phase?
5. Is it necessary to have forward secrecy (in the case of disenrollment, is it necessary to prevent disenrolled shareholders from retaining potential access to future secrets)?
7. Is it necessary to have backward secrecy (in the case of enrollment, is it necessary to prevent new shareholders from obtaining access to past secrets)?
8. Is it necessary to inform shareholders after an update operation has taken place and, if so, how is this done?

D. Security design issues

A number of design issues need to be considered that explicitly relate to the security design of the scheme.

1. Does the secret need to be explicitly or implicitly reconstructed (see Figure 1) by the combiner? (This is related to the trust assumptions concerning the combiner).
2. Are there restrictions on the cryptographic mechanisms that can be employed (for example, are certain algorithms already employed elsewhere in the system that could be reused)?
3. What security model does the resulting scheme need to offer security in (for example is unconditional security required, or is computational security sufficient)?
4. What threat model applies to the scheme (for example regarding likelihood of compromise of entities)?
5. What level of robustness is required against malpractice by any entities involved in the scheme? This could include:
   - Accidental damage or loss of shares;
   - Deliberate manipulation of shares;
   - Fraudulent combining of shares.

E. Resource constraints

Finally we identify a range of questions that are important with regard to the ability of entities to successfully carry out operations.

1. What constraints are there on the amount of data that entities can store securely?

2. What constraints are there on the computational capability of entities within the scheme?
3. What bandwidth restrictions are there on the communication channels between entities?
4. What priority should be placed on reducing connection complexity between entities?

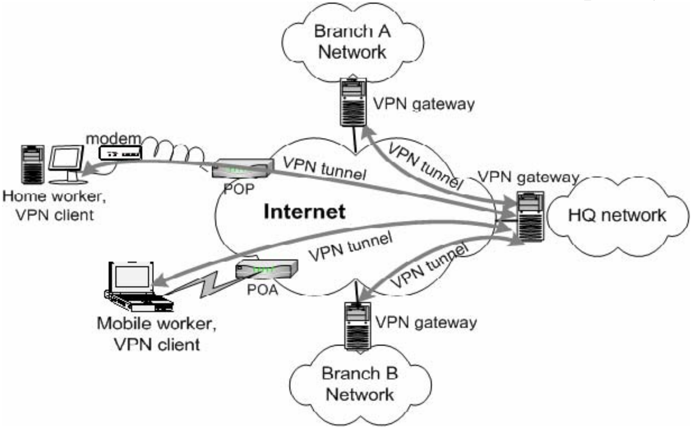


Figure 2: VPN Network: secure tunnels between branch A's and HQ's gateway, branch B's and HQ's gateway, home worker and HQ's gateway, and mobile worker and HQ's gateway. Networks beyond the VPN gateways are protected networks.

The sets of issues identified in this section are, to varying extents, interdependent. What it is most important to recognize however is that it is not possible to design a practical solution without careful consideration of all of them.

6. SCHEME MANAGEMENT WITHIN VPNs

To demonstrate the utility of our framework, we will show how to apply it to a sample environment, which in this case is a generic Virtual Private Network (VPN) [23] application. Note that for brevity we necessarily study this application at a relatively high level, which is quite acceptable as it is being used to demonstrate efficacy of the framework and not to specify a complete solution.

First, we present the topology of a VPN supporting mobility or remote-access capability of shareholders and then we use the framework to propose a security blueprint.

A. Virtual Private Networks

A VPN is a way to simulate a private network over a public one, such as the Internet. It is called ”virtual” because it depends on the use of virtual connections that have no real physical presence. Secure virtual connections can be created between two machines, a machine and a network, or two networks. Secure virtual connections are constructed using tunneling protocols (e.g. PPTP [24], L2F [25], L2TP [26] and IPsec [27]) which offer, in addition to the tunneling functionality, many security services such as authentication and confidentiality. Figure 2 illustrates the case of a VPN consisting of three local networks of the head quarter, HQ, and two branches, as well as a remote access home worker and mobile agent (entities using IP mobility).

In general, mobile agents use mobility technology or mobile IP [28], which allows them to connect to their home networks or to other corresponding entities (mobile or fixed) via secure connections. This means that they use two types of tunneling: mobile IP tunnels over VPN tunnels and thus they offer two layers of security.

B. Threshold cryptography with a VPN

1. Scheme management model

Many different types of operation (such as a group signature, dynamic voting or bank transaction approval) might need to be jointly carried out by a predetermined set of entities (shareholders) within such a VPN scenario. Those entities are either within their own network, which is a part of the VPN, home workers, or mobile agents. They can thus be considered to be within a relatively controlled and secure environment. Further, in such applications, security tasks and operations are typically controlled by a security-specialist agent. This agent is thus a prime candidate for the role of dealer and should be made responsible for the threshold scheme management. Information about these operations could be dictated by a higher responsible entity, conforming to a pre-defined security policy. As a result the scheme management model should be autocratic; the dealer is entirely responsible for scheme management and there is no need to communication between shareholders. For these reasons VPN application environments belongs to Class B. If we consider that the VPN environment to be secure then there is no need to further secure links between the dealer and shareholders, thus scheme management models 6 is the most suited. If links within the VPN cannot be assumed to be secure then model 3 is best suited to be the scheme management model.

2. Roles and responsibility

In most companies, people participating in critical operations are determined by a top-level administrative entity.

The dealer must be made aware of the list of shareholders and the minimum number participating in a threshold operation. Further, in the selected management models the dealer is responsible for the scheme management operations during both the setup and running phases. The dealer creates and securely distributes shares to participants. Since most applications are likely to be relatively long-lasting, the dealer should be able to update secret keys and refresh shares. In addition, the dealer will be responsible for update operations such as adding new participants and evicting cheating or retiring shareholders.

Shareholders should have the ability to verify shares on receipt. Shareholders are responsible for maintaining their shares and producing partial results that are sent to a combiner that acts on behalf of the group as the virtual signer or decision maker.

Similarly to the dealer, the combiner should be designated in a way that he/she is close to the administration. It is important that the dealer and combiner should not be nomads.

3. Networking environment

During the initialisation phase there will be a need for unicast and multicast connections between the dealer and the shareholders. The unicast connections are used to distribute shares and should be secure (authentication and confidentiality). An ASM (Any Source) multicast capability amongst the dealer and shareholders would allow any shareholder to inform other shareholders about anomalies such as incorrect shares or a compromised dealer. This would also allow the dealer to diffuse verification commitments or witnesses prior to share distribution. During the running phase, multicast between dealer and shareholders will be used to refresh the secret key and shares. Unicast links are needed between the combiner and the shareholders during secret reconstruction.

In both phases, all unicast links must be secure. However, this will have only a minor impact on the network performance. This is because links between any two scheme entities are formed of two parts: external (between two VPN gateways or between a remote worker or mobile agents and a VPN gateway) and internal (inside a VPN domain). The external parts are secure because they are formed over VPN or mobility tunnels. Security of the internal parts is not a real concern because they are intra-domain (inside the domains).

4. Update operations

In the type of application being considered almost all types of update might be required. For example shareholders might relinquish their role, others might join the group, changes to the threshold (augmenting or diminishing the number of the signing shareholders) might also be considered during the running phase. These changes must be reflected at the scheme management level. The dealer must be responsible for conducting these operations and they not be limited in number. The need for updates should be evident on initialisation, but beyond that the precise nature of updates is unlikely to be able to be specified in advance. Furthermore, for administration reasons, it is not necessary to inform other scheme entities (save for the combiner) about any change. Forward secrecy will only be required if the combiner is unable to enforce it by refusing partial results produced by expelled or retired shareholders. (A similar remark applies to backward secrecy).

5. Security design issues

Most of the envisaged applications (e.g. secure signing) are likely to involve implicit secret reconstruction. There are unlikely to be significant restrictions on cryptographic mechanisms as most secure tunneling standards support cryptographic primitive negotiation. Computational security is likely to be sufficient. It is likely that protocols will need to incorporate robustness against manipulation of shares by shareholders. The dealer and combiner can probably be trusted to perform their roles honestly.

6. Resource constraints

Since we assume that the combiner and dealer are not mobile and that shareholders do not participate in scheme management except using their shares, the only significant constraints apply to shareholders who operate as mobile agents, with relatively limited computational and communication resources. It may thus be judicious to select a technique that shifts significant computation onto the combiner.

7. CONCLUSION

We introduced in this paper an abstract framework for managing threshold cryptography in distributed network environments. We built our framework around a classification of management models, application classes and sets of security issues that need to be considered. Further, we instantiated the framework within a VPN environment to generate an architectural blueprint for designing a suitable system. We stress that the focus of this paper is on taking a holistic approach to identifying the requirements for implementing threshold cryptography and does not provide a taxonomy of technical solutions. Rather it helps to clarify the relevant parameters within which candidate solutions can be suitably evaluated. Future work should be directed at using this framework to assist in the design of solutions for specific applications.

9- GLOSSARY

Authentication	وثوقية
Architecture	بنية
Blueprint	مخطط
Combiner	مُركِب
Confidentiality	سرية
Design issues	مشاكل تصميمية
Digital signature	توقيع رقمي
Distributed environment	بيئة موزعة
Disenrollment	إقصاء
Dealer	وكيل، موزع
Encryption	تشفير
Enrollment	انضمام، التحاق
Framework	إطار عمل
Implementation	تنفيذ
Initialisation phase	مرحلة تهيئة
Key	مفتاح
Management	إدارة
Model	نموذج
Multicast	بث متعدد الوجهات
Proactive	استباقي
Refreshment	تحديث، تجديد
Running phase	مرحلة تشغيل
Share	حصة
Shareholder	مساهم، مشارك
Security mechanism	آلية أمنية
Secret sharing	مشاركة سر
Threshold scheme	مخطط عتبي
Threshold cryptography	التعمية العتبية
Tunnelling protocol	بروتوكول نفق
Unicast	بث وحيد الوجهة
Update	تحديث
Virtual	افتراضي

8-REFERENCES[*]

[1] Y. Desmedt and Y. Frankel. "Threshold Cryptosystems," *Advances in Cryptology: Crypto '89, Lecture Notes in Computer Science*, 435 (1990) 307–315.

[2] B. Blakley, "Safeguarding Cryptographic Keys", *Proceedings AFIPS 1979 National Computer Conference*, pp. 313-317. June 1979.

[3] A. Shamir. "How to share a secret", "*Comm. ACM*, 22(11), (1979), pp. 612-613. November 1979.

[4] E. Brickell, G. Di Crescenzo and Y. Frankel. "Sharing Block Ciphers", *Information Security and Privacy, Lecture Notes in Computer Science*, 1841 (2000) 457–470.

[5] R. Canetti and S. Goldwasser. "An Efficient Threshold Public-Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack," *Advances in Cryptology: Eurocrypt'99, Lecture Notes in Computer Science*, 1592 (1999) 90–106.

[6] H Liu, W. Xie , J. Yu , P. Zhang , and S. Liu. "A general threshold encryption scheme based on new secret sharing measure," *Proceedings of ICIEA'11*, Beijing, 2011.

[7] D. Chaum and E. van Heyst. "Group Signatures". *Advances in Cryptology: Eurocrypt '91, Lecture Notes in Computer Science*, 547 (1991) 257–265.

[8] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. "Robust Threshold DSS Signatures," *Advances in Cryptology: Eurocrypt '96, Lecture Notes in Computer Science*, 1070 (1996) 354–371.

[9] C. Park and K. Kurosawa, "New ElGamal Type Threshold Digital Signature Scheme", *IEICE: IEICE Transactions on ommunications/Electronics/ Information and Systems*, 1996.

[10] D Gordon, J. Katz, and V. Vaikuntanathan. "A Group Signature Scheme from Lattice Assumptions," *Proceedings of Asiacrypt '10*, Singapore 2010.

[11] D. Boneh and M. Franklin. "Efficient Generation of Shared RSA Keys," *Advances in Cryptology: Crypto '97, Lecture Notes in Computer Science*, 1233 (1997) 425–439.

[12] P. Feldman. "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", *Proceedings of. FOCS'87*, PP. 427- 438. 1987.

[13] L. Harn and C. Lin. " Strong (*n*, *t*, *n*) verifiable secret sharing scheme," *Information Sciences*, 180 (2010) 3059–3064.

[14] D. A. Schultz. "Mobile Proactive Secret Sharing," *Master Thesis*, MIT, 2007.

[15] , A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. "Proavtive Secret Sharing or: How to Cope With Perpetual Leakage", "*LNCS 963, Proc. Crypto'95*, Springer Verlag, pp. 339–352. November 1995".

[16] L. Baiand X. Zou. "A Proactive Secret Sharing Scheme in matrix projection method," *Int. J. Security and Networks*, Vol. 4, No. 4, pp.201–209.

[17] S.G. Barwick, W.-A. Jackson, and K.M. Martin, "Updating the parameters of a threshold scheme by minimal broadcast," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp.620–633, 2005.

[18] B. Blakley, G.R. Blakley, A. Chan, and J. Massey, "Threshold schemes with disenrollment," *Adv. in Cryptology - CRYPTO'92, Lecture Notes in Comput. Sci.*, 740, pp.540–548, 1993.

[19] C. Blundo, A. Cresti, A. De Santis and U. Vaccaro. "Fully dynamic secret sharing schemes," *Adv. in Cryptology - CRYPTO'93, Lecture Notes in Comput. Sci.*, 773, pp.110–125, 1994.

[20] M. Li and R. Poovendran. "Broadcast-enforced disenrollment in threshold schemes," *SAC 2003, Lecture Notes in Comput. Sci.*, 3006, pp.101–116, 2004.

[21] K.M. Martin, R. Safavi-Naini, and H. Wang. "Bounds and techniques for efficient redistribution of secret shares to new access structures," *The Computer Journal*, vol.42, no.8, pp.638–649, 1999.

[22] R. Steinfeld, J. Pieprzyk and H. Wang. "Lattice-based threshold-changeability for standard Shamir secret sharing schemes," *Adv. in Cryptology – Asiacrypt'04, Lecture Notes in Comput. Sci.*, 3329, pp.170–186, 2004.

[23] C. Scott, P. Wolfe, and M. Erwin."Virtual Private Networks", *O'Reilly, 2nd Edition*, 1999.

[24] K. Hamzeh, et. *al.*, "Point-to-Point Tunneling Protocol (PPTP)", RFC 2637, IETF. July 1999.

[25] A. Valencia, M. Littlewood, and T. Kolar. "Cisco Layer Two Forwarding Protocol (L2F)", RFC 2341, IETF, May 1998.

[26] W. Townsley, et *al.,* "Layer Two Tunneling Protocol (L2TP)", RFC 2661, IETF. August 1999. [27] R. Atkinson and S. Kent, "Security Architecture for the Internet Protocol", RFC 2401, IETF. November 1998.

[28] C. Perkins. "IP Mobility Support", RFC 2002, IETF, October, 1996.

[*] Received 22/11/2012.