Assumptions Made for the British Telecom URI Project

Paul Wernick

17th July 1995

'Robust security designs are those that make their assumptions explicit' ([2, p.40])

1 Management Summary

This document describes in detail the assumptions which are to be made in defining the security architecture for the URI project. This work is important in that it helps to set the limits to the architecture's scope, gives a basis for informed critiques of the architecture, and provides the designers of implementations based on it with a series of rationales for making design decisions in the light of knowledge of the architecture's objectives. It will also provide a basis for modifying the architecture in a controlled manner should any of these assumptions become invalid over time due to experience or changes in the environment within which designs based on it are operating.

This document, and the assumptions presented in it, are to be regarded as provisional conclusions only. As the project advances and knowledge of the environment within which implementations of the architecture will have to operate and the constraints placed on such implementations, these assumptions will need to be re-assessed either continuously or on a regular basis.

Our assumptions have diverged in many ways from those in two documents prepared by British Telecom staff for the TINA consortium, [6] and [7]. The reasons for this can be broadly summarised as being due to our desire for independence for the architecture of all design and implementation technical matters, our taking a wider view in identifying the stakeholders in the architecture both with regard to their identities and their degree of sophistication in security-related matters, our belief that access to the services protected by the architecture must be as easy as possible to make it more likely that they will be used, and our assumption that not all of the entities involved in the network will trust each other.

We hope that the completed architecture will have a long lifetime compared with specific aspects of the environment within which it is to be used, and therefore believe that it must remain independent of any specific technical, security or commercial assumptions or interests. This fundamental concern underlies many of the assumptions described below.

2 Introduction

2.1 Setting the Context

This document describes the assumptions which are to be made concerning the URI architecture and the world within which it is intended to operate. These assumptions will constrain the design of the architecture, and feed into the functional and non-functional requirements of the architecture to be designed.

The information is set out in the form of a initial set of assumptions regarding the nature and extent of 'architectures', followed by a series of assumptions as to the stakeholders in the architecture and the boundaries of the architecture and some general beliefs and assumptions.

The assumptions related to the detailed environment of and interactions with the architecture are divided into those which relate to the technical environment, the security environment, the legal and regulatory environment, the values implicit in any statements of the architecture, the threats against which systems developed to conform with the architecture will need to defend themselves, and relevant aspects specific to service providers, service users and network carriers.

Finally, outstanding matters are considered in terms of the assumptions still needing to be made or considered, the people to whom it may be worthwhile referring in extending the assumptions requiring to be made, and a number of caveats concerning the direction of this document.

The burden of building implementations to meet requirements framed to meet these assumptions may be considerable at the current state of technology. It may therefore be necessary to consider whether subset implementations meeting the architecture are to be permitted, and if so whether these are to be formalised into a series of *levels* of compliance with the architecture.

2.2 Sources of These Assumptions

The assumptions described below have been obtained or derived from the following sources:

- our own knowledge, experience and introspection,
- work by other researchers at the Computer Laboratory,[1]
- consideration of literature concerning authentication in networks, such as [13], and
- a critical examination of two documents describing work for the TINA consortium relevant to this project, [6] and [7].

In some cases, the assumptions presented here have been contrasted with assumptions in published work in order to illuminate the differences between them.

2.3 Caveats

There are some remaining concerns regarding the contents of this document. These include the following:

Conflicting Assumptions It will be noted that some of the assumptions listed below seem to contradict others. This is on some occasions the result of conflicts between the requirements of the stakeholders, and elsewhere due to the conflicting requirements of a single stakeholder.

For example, a service user might want access to a service to be as easy as possible (so that it is easy to use) and as secure as possible (so that nobody else can use the service at the user's expense), requirements which may conflict and therefore need to be traded off against each other.

Resolving these clashes will form a part of the next step in the URI research.

Content of this Document This document currently intermingles assumptions, notes on the scope, and thoughts on the requirements of the architecture. These need to be divided, probably by separating the following for each assumption, possibly in the form of a table or by the use of sub-headings: what it is, what its effect is on the scope of the project, and first thoughts as to its possible or potential influence on the requirement of the architecture.

This work is to commence shortly.

[1] In particular, the threat model being developed by Shaw Chuang.

3 The Nature of Architectures

The types of components to be found in a possible URI architecture are described here. These will form the basis of what can, and ought, to be delivered as the project's products.

3.1 Form of an Architecture

In his description of the ANSA architecture, Herbert complains that 'Most other 'architectures' known in the computer and telecommunications industries have a fixed structure similar to a systems design.' ([5, p.19]). He regards the ANSA architecture as a mechanism for designing, rather than a design. It how things should be done, rather than exactly what to do. Examples of software systems developed using the architecture are therefore variable in form, content and purpose, but all can inter-work and reflect the same underlying philosophical base.

We suggest that we regard the URI architecture in the same way, as closer to a methodology than a design.

3.2 Content of an Architecture

Herbert ([5, p.19]) states that the ANSA 'architecture' consists of four types of elements, viz.

- a set of *components* – the building blocks and tools of the architecture,
- a set of *rules* for combining components,
- a set of advisory *recipes* for combining components into systems with certain properties, and
- a set of *guidelines* to assist designers.

We suggest that this is a good starting point for what the URI architecture should (and should not) contain.

Additionally, we believe that a number of published detailed exemplars of good practice in using the architecture would be of value to those who might be new to the architecture, or unsure as to how to tackle a particular problem or issue in the spirit of the architecture.

4 Stakeholders in the Architecture

In order to determine the requirements of the architecture, and the constraints within which it might operate, it is necessary to know who has a stake in the architecture, at least to the extent that they can influence the design of systems based on that architecture. Potential stakeholders in the URI architecture are identified here; some of the fine detail of these subdivisions may in time turn out not to be useful. The aim is to describe those persons involved in the use of the architecture whose different requirements, rights or obligations may result in the need for different services from or or interactions with systems designed under the architecture.

- service providers:

  content providers: who provide on a wholesale basis material controlled by their copyright for use in service provision; this may include data (such as searchable databases, prerecorded films or live broadcast coverage) and/or program material such as games software or systems to provide the functionality to make better use of data,

  service retailers: who sell services to users, and

information or service use brokers and advisors: who act as middle-men between service providers and users; this subgroup may for instance include information scientists and librarians who provide a value-added service advising users on the quality of data in particular databases or perform sub-contracted searches; this class may be considered to be both service providers and service users, and may eventually be subsumed into one or other of these groups;

- service users:

  customers: those responsible for paying the bills for network and/or service use, and

  users: who actually make direct use of the services via the network;[2]

- network carriers:

  network owners: who own the physical assets comprising the networks, and

  network operators: who manage and operate the networks on behalf of the network owners;

- security system providers:

  security architecture owners: those who own the intellectual property rights in the security architecture,

  security architecture developers: those who originally developed the security architecture,

  security architecture maintainers: those charged with maintaining the security architecture up-to-date and relevant,

  security implementation owners: those who own the intellectual property in any particular security system implementing the architecture,

  security implementation developers: those who originally developed any particular implementation of the security architecture,

  security implementation maintainers: those charged with maintaining any particular security architecture up-to-date and relevant, and

  security managers: who maintain the data structures and facilities supporting authorisation and authentication mechanisms; these people may be employed by other stakeholders but their position of power over the security mechanisms may imply wider responsibilities than those imposed by their employment,

  unless the security systems can be regarded as a specific *service*, which would place the above as service providers of different sorts;

- the wider society: as evidenced by

  formalised by legal and regulatory bodies: to whose requirements the other stakeholders are assumed to comply under all but the most extreme circumstances,

  informal pressures: which might for instance be political or commercial; this group is not recognised in [6] and [7] as a valid stakeholder, nor as a body whose influence needs to be considered at all, instead relying on regulatory and statutory bodies to channel such pressures when they become sufficiently important, and

  society as a set of individuals: with rights, such as as those to privacy and data protection, and obligations, even if they are not service users or providers and such protections are only weakly supported by legal and regulatory requirements.

[2] Indirect users of information gained from service are not included here.

Additionally, a requirement for a 'trusted party' in certain authentication mechanisms may result from architecture-based implementation decisions.

The majority of the analysis which follows in this document currently only recognises the top-level division of stakeholders; i.e. service providers, service users, network carriers and wider society.

Service providers, service users and network carriers can be subdivided into *current* and *potential*, to capture the assumption that the burdens of the architecture must not be perceived by potential users, providers and carriers as being sufficiently onerous to dissuade them from creating, using or carrying services.

As will be seen below, the influence of wider society is important in considering the architecture, since it will be necessary to operate within legal and regulatory frameworks at least nominally required by that society.

Assumptions concerning the stakeholders are:

- Service providers and network carriers are assumed to be corporate bodies. Service users might be either corporate bodies or private persons.[3]

- The ways in which systems conforming to the architecture may be designed or used make it impossible to assume that the network carriers are the same as either the service providers or the end users. Therefore these three – service providers, network carriers, and service users – should be regarded as separate for architectural purposes, despite the likelihood that any one might at times act as either of the others or in all three roles.

  This assumption will make it easier to examine the effect of legal and regulatory measures on the architecture.

- Any trusted party required by an implementation of the architecture will not necessarily be associated with any of the stakeholders in any way, other than a presumed requirement to conform to laws and regulations. If it exists, the trusted party may be considered to be an additional stakeholder, and analysis of the situation should include a consideration of the costs and benefits to the trusted party of conforming or not conforming to expected behaviour.

- 'Society' will in the first instance be assumed to be the U.K., with additional influence from the European Union and the U.S.A. as expressed in U.K. treaty obligations.

- The stakeholding of 'society' will be represented through the medium of laws and/or regulations which will be imposed from outside (possibly after consultation), and codes of conduct agreed between the other stakeholders. Any or all of these laws, regulations and codes may change from time to time in response to actions either related to the architecture, such as perceived weaknesses, or unrelated to the architecture, such as media hysteria.

  No other activities by persons or organisations representing 'society' are to be explicitly supported in the architecture.

- No assumptions are made about the internal structure, operational mechanisms and so on of any trusted parties. This is because the existence of such parties is not a requirement of the architecture and its records cannot be therefore relied upon to exist in all implementations under the architecture. This assumption also allows designs within the architecture to regard any such trusted function as a black box, which will aid its replacement by another mechanism if required for security purposes.[4]

[3] The distinction between corporations and private people becomes significant when the different treatment of the two by regulatory regimes is considered.

[4] Note that the architecture requires any security-related mechanism to be replaceable by another in order to maintain security as time passes.

To restate the assumption, any details of the implementation of trusted parties are to be considered as being outside the scope of the architecture. Should one or more of them be needed for a particular implementation, their external properties can be used in architecture-level designs, such as taking advantage of the existence of trusted sources of authentication data, but the internal designs and the mechanisms which make them trustable are to be considered at a lower level, i.e. nearer to the implementation, than that of the architecture.

5 Boundaries of the Architecture

The boundaries of the architecture are described here, in terms of a set of broad assumptions as to what is to be included in and excluded from the architecture. In addition to the general architecture boundaries given here, some of the more detailed assumptions have explicit in them boundaries for the specific issues addressed,with the result that some of these boundaries are re-expressed in more detail below.

5.1 What is to be Included in the Architecture

- The purpose of the architecture is to provide authentication and authorisation services in an arbitrary network, subject to the further assumptions listed below.
- The architecture is to provide functionality to form and break connections between service users and services, according to a set of security policies which are to be defined outside the architecture but which must be implemented by it.
- The architecture is to make general statements concerning the computer-based systems providing the required functionality. It will additionally provide, as a minimum, advice on the requirements of administrative functions to support the required level of security, and will provide tools such as notations and guidelines to support the design by implementers of these necessary off-line procedures.
- Some functional requirements described in the architecture may be optional in implementations conforming to the architecture. An example of this is a trusted time service, which may be needed for such aspects as trustable time-based charging (whether connect time or time of day) or time-based access controls to services.

5.2 What is to be Excluded from the Architecture

- Any assumptions concerning the nature and content of the security policies to be implemented under it are outside the scope of the architecture. The architecture must be sufficiently flexible to permit the implementation of any reasonable, foreseeable security policies.
- Any requirement to provide support for the examination of data messages[5] passed between users and services between user-service connection and disconnection will be outside the scope of the architecture.
- Any record-keeping required after any authentication between the service user and service provider has been completed and both sides have agreed where applicable on the set of rights allowed to the user over use of the service are outside the boundary of the architecture. However the architecture will not prevent any such record-keeping by any parties involved in a connection.

[5] i.e. those unconnected with the functional and non-functional requirements of the architecture

- There will be no requirement to provide facilities in the architecture for any examination by third parties of the transmissions and records maintained or augmented by the architecture. Any such examination of transmissions and/or records is outside the scope of the architecture.

- The initial gaining of access to the network by service providers and service users is outside the scope of this architecture. This does not preclude some network security procedures after the mechanical access to the network has been gained.

- The design and implementation of specific encryption algorithms and similar low-level authentication protocols are outside the scope of the architecture. The architecture should however allow as much freedom as possible in selecting and replacing such mechanisms of these types as are required by any particular implementation of the architecture.

- The architecture will not be concerned with or reflect any commercial pressures on any of the stakeholders, other than a general concern that its implementation be not so expensive as to prevent its being usable in the real world.[6]

6 General Beliefs Implicit in the Architecture

Beliefs which are implicit in the architecture, but more specific to it than general statements about developing software, include assumptions that the following are good things:

- minimising the size of the Trusted Computing Base,

- minimising the number and importance of bottlenecks in systems systems developed to meet the architecture,

- an appropriate level of and mechanism for security for each service protected by systems developed to meet the architecture.

7 The Technical Environment

The assumptions to be made about the technical environment within which the architecture will have to operate include the following.

- The architecture will make no assumptions concerning the implementation of the carrier's network, other than that it will be a network capable of carrying digital information connected by means outside the architecture to both service users and service providers, and to any trusted parties required by an implementation.

- The architecture makes no assumptions as to how the architecture will be implemented, in particular whether any particular part of the implementation will be in hardware or software, where the hardware/software boundary might come, and whether the implementation will be based on shared use of general-purpose computing facilities or on dedicated equipment (cf. [11, p.2-3]).

  However the architecture will not of itself prevent any technically feasible division of implementation between hardware and software, or the use of resources shared with other applications. Decisions as to whether and to what extent specific shared resources can be trusted are outside the scope of this architecture.

[6] This is a point of difference with the attitude of the TINA documents.

- It will not be possible to foresee all of the possible implementation mechanisms for systems designed under the architecture (cf. [6, p.8]). The architecture must be capable of simple modification to meet new implementation models with the minimum of change to existing implementations. It is also assumed that some aspects at least of the architecture will need to be implemented in existing systems, designed and implemented using older notations and associated computational models

  This may imply that if the components of the architecture are described using a particular computational model, for example the object-oriented model, guidance should be given to implementers using tools based on other computational models.[7]

- The operations and interactions of the service users, carriers and service providers may (or may not) be distributed arbitrarily throughout the digital network.

- The components forming the security architecture may (or may not) be distributed arbitrarily throughout the digital network.

- The technical world will be heterogeneous, with a variety of protocols, hardware and software environments, networking mechanisms and so on. The boundaries thus formed may or may not coincide with political, geographical or economic boundaries of security sub-domains.

  This technical world will be in a constant state of flux, with new mechanisms arriving as at least the same rate as old ones are discarded. This situation will not change within the lifetime of the architecture. However, inter-working at the digital level will be assumed to be maintained, and the mechanisms at the relevant levels are therefore outside the scope of the architecture. At the level at which it will operate, and in its interactions with higher and lower layers, the architecture should be as portable as possible, and insulated as far as possible against technical changes such as those described here, as well as against changes in design computational models and implementation languages.

- Some technical standards in matters related to the architecture will become generally accepted over time, and these should be adhered to where possible and where they can be recognised as being of use. This position should be maintained at the expense of non-standard optimisations of the architectural components.

- Low-level technical issues related to message transmission, such as the disassembly, transmission and reassembly of messages and flow control, are outside the scope of the architecture. It is assumed that the network will be able to perform these activities routinely, and generally (but not necessarily always) perform them correctly.[8]

- No assumption is made that a single connection between two parties is other than a unidirectional single channel, i.e.. no out-of-band or bidirectional facilities are assumed to be present and available to the architecture. This is to ensure compatibility between the architecture and those networking mechanisms which do not provide such facilities.

  It should be noted that a 'back channel' or bidirectional working between two parties can be implemented by means of a two unidirectional channels, albeit usually at some additional cost.

- The complete, uncorrupted delivery of architecture-related messages sent through the digital network over which the operations of service providers and users are distributed cannot be guaranteed by any mechanisms provided outside the architecture. A different message may

[7] For a parallel to this, and as an example to be adopted if the architecture is described in object-oriented terms, see the advice given to implementers of OMT designs in non-object-oriented languages and database management systems by Rumbaugh *et al.* ([16, chapters 16-17]).

[8] This assumption immediately places us fairly high up the ISO 7-layer mode, at layer 4 (the transport layer) or higher. It probably also constrains us to the consideration of end-to-end rather than link-based security mechanisms (cf. [17].

arrive than the one which was sent, due to malicious intervention, accident or equipment or software failure.

The architecture will therefore need to protect itself against such corruption of messages. Error detection (and optionally correction) at the message level is within the scope of the architecture; network-level error correction is outside the scope of the architecture.

The delivery of data messages is outside the scope of the architecture.

- The complete and error-free delivery of architecture-related messages sent through the digital network over which the operations of service providers and users are distributed without unauthorised access or modification cannot be guaranteed by any mechanisms provided outside the architecture. Since the state of the network will be such that the completion of a multi-message interaction between a service user and a service provider cannot be guaranteed, the architecture will have to guard against failures during its interactions.

  Additionally some mechanisms will be required to allow for the situation in which calls in progress drop out after completing the authentication and authorisation procedures provided by the architecture, to prevent a third party either gaining access to the already-authenticated service connection or impersonating the service. This might require some interaction with the network, say to receive a message that the connection has failed and update the architecture's records accordingly. However, this should be implemented in as portable a manner as possible.

- The mechanisms provided in the network communication infrastructure to support message routing and entity identity cannot always be relied upon to provide the degree of protection required of a security architecture such as this.

- Technical solutions of sufficient security and robustness will be available to the problems of authentication and authorisation, and the best of these solutions will be, and remain, sufficiently in advance of attempts to circumvent them to allow their use in implementations based on the architecture.

  This statement might seem like a requirement of the architecture reduced to an assumptions, but without this underlying belief that the problem can be solved, there is no point in developing the architecture at all!

- Any program is likely to have bugs in it. This includes those programs supporting the services and their accounting mechanisms, as well as those implementing the architecture itself.

  The design of the architecture must minimise any harmful effects of such bugs, particularly where they might affect the commercial or legal position. For example, an occasional failure of service may be considered to be less important than a hole in the billing mechanism which prevents charges being raised or renders them uncollectable.

  Additionally, manual and/or computer-based mechanisms should be provided to check for the correct operation of the security-related features of systems, which do not assume that the systems generally run correctly i.e. they should be looking for accidental as well as deliberate damage.

- The network infrastructure will have sufficient capacity to provide for the transmission and receipt of any messages which are to be sent in implementing the architecture. However this assumption is not intended to suggest that efficient network capacity use is outside the scope of the architecture (see below).

- It is assumed that use of the network will carry some 'cost' to the effect that the protocols implementing the architecture's security mechanisms should not use network resources wastefully. However, no assumptions are made in the architecture as to what is the most efficient way of using the network, in order to accommodate different networking technologies as they arise. Essentially, such optimisations are left to implementers of designs under the architecture.

To summarise, the technical environment will be unstandardised and becoming more so, with the exception of a few generally-accepted standards, and the network will be digital, insecure and unreliable.

8 The Security Environment

Necessary and/or useful assumptions about the security environment are listed below.

- It cannot be stated with certainty that any set of security mechanisms will detect all deliberate or accidental security breaches. In addition, the security system is finally dependent on the people who operate it. Audit trails and logs of actions, which cannot be modified (and preferably not read) by those whose actions are logged, and which are examined regularly for exceptional activities requiring further investigation, are therefore essential.
- A service user, service provider or network carrier should not have to rely on any element of the architecture outside his or her security domain for the security of his or her data, unless that element is specifically designated as 'trusted' (cf. [4, p.477]).
- The participants in the use of the architecture, i.e. service providers, service users and network carriers, will not necessarily trust each other. This lack of trust extends to elements of authentication protocols such as message sequencing codes.

  The architecture will need to provide control and recording facilities, to allow the use of services in the absence of trust. The assumption may also restrict the range of security mechanisms which can be used.
- It is assumed that for security purposes implementations under the architecture can only differentiate between those terminals *securely* connected to a secure infrastructure, such as a terminal connected by a direct physical link to a trusted server and locked in the same room or a smart-card encryption-protected link, and other *less secure* terminals. There may be a need for a variety of different security degrees assigned, for instance, to a particular combination of a human person or a delegated program at a predefined location, or a person at any location, etc. This set of degrees of security will not necessarily form a single well-ordered hierarchy. The assignment of people, terminals and links to different degrees of trustworthiness is a matter for security policy and therefore outside the scope of the architecture. It will ultimately have to be implemented and administered by humans or delegated programs. However the architecture should support a reasonable range of possible security degrees to make the work of providing any required differentiation.

  The use of network addresses outside a domain of security as a source of information for authentication and/or authorisation puts the provider of those addresses within the relevant security domain, and, since by earlier definition the parties involved in operating and using the architecture's implementations do not trust each other this is not an acceptable solution.

  It should also be noted that any information gained by a program running on a computer concerning whether a terminal is directly connected or not relies on trusting both the software and hardware within the computer supplying the physical addresses reflecting the port to which the terminal is attached, and the integrity of the physical connection from the port to the terminal, neither of which can be checked by the program itself. This consideration, generalised as a security concern, may militate in favour of the use of certified secure computer systems to host the security architecture in order to provide some comfort as to the security of the host system hardware and its systems software. However, it is not proposed to include the trusted hardware manufacturers and systems software authors as stakeholders in the system to reflect their input to the security of the implementation.

- The security requirements of network administrators are not sufficiently different from those of service providers and users to require the provision of a completely different security environment. Common mechanisms can be devised which meet the needs of network administration and operation and those of service users and providers, at least at the higher levels of abstraction.

  This assumption does not exclude the possibility of the need for considerable differences between the implementations of security mechanisms for network administrators and for service providers and users; special provisions may be required, for example, in the case of a network administrator starting a network switch from cold.

- An unambiguous mechanism, developed outside the architecture, will be available for identifying services to users. The current telephone system suffers from premium numbers being sold and re-sold, so that users find an unexpected service at the (apparently) correct number. The link of number to service is transitory and insufficiently stable to support the environment envisaged under the architecture.

  It is assumed that a better way of identifying services protected by systems developed under the architecture will be found, and that providing assurance to service users that they will get through to the requested service or not be charged will form part of the architecture

  This assumption may lead us towards a secure, guaranteed-unique entity naming system – see [7, p.14], where the need for a secure naming scheme is set down as a sub-requirement of the major objective of identifying, authenticating and authorising all entities.

- 'Users must be responsible for their own security.' ([4, p.477])

  We assume this, at least to some extent. If a user is sufficiently careless, then almost any security system will break down eventually. The mechanisms which eliminate this aspect of security, for example by using users' physical metrics as identifiers, are currently too expensive to use in a consumer environment, and it is assumed that this will not necessarily change in the future.[9]

  However, in contrast to the environment which Denning envisaged of users being sufficiently technically aware to select appropriate security safeguards, it is not assumed here that all users or the facilities provided by the architecture will be sufficiently knowledgeable to do so. Service providers and network carriers may know enough to decided on their own security levels (although some of the less sophisticated service providers might not), but personal service users cannot be assumed to be aware of the threats and possible countermeasures and they must be supported in the architecture without automatic reference to their own understanding of the situation, probably by the provision of a small number of default security levels as well as a customisable environment.[10]

  It will be assumed that network administrators will be aware of security considerations related to their work, and will be responsible for their own and delegated security-related work.

- It will not be possible to foresee all of the possible threats to systems designed under the architecture (cf. [6, p.8]). The architecture must be capable of simple modification to meet new threats with the minimum of change to existing implementations.

[9] If this situation does change, and such secret-free identification becomes feasible, the architecture should be able to take advantage of such technologies.

[10] It may be necessary to give some warning to users before they customise the security environment, to prevent the architecture's implementers becoming liable for errors by the customiser. For similar reasons, the features and names of the default security environments should be chosen with care to prevent over-reliance by misguided users.

- It is possible to encapsulate generalised authentication and authorisation models in one generic framework.[11]
- The architecture will make no assumptions as to whether any parts of the mechanisms defined in designs based on it and external to the architecture-based components can be trusted. This decision will be left explicitly to implementers.

  However the level of trust assumed in any particular design will not be greater than the maximum possible under the other constraints in the architecture, such as legal or regulatory requirements.
- Any single instance of a trusted element (such as a server, algorithm, key, or person) of a design developed under the architecture might be compromised, and therefore become untrustable, at any time. Generic trusted elements, such as roles, will remain trustable despite security compromises of this type.
- Mechanisms for multiple or shared authentication may be required in addition to single entity authentication.
- Any mechanism designed to improve security in a system will be broken sooner or later, whether by benign or malicious attackers. In addition, a security-related mechanism's vulnerability will increase over time, as means of breaking it improve and computational facilities become faster and cheaper.

  The architecture should enable the easy replacement of one security mechanism by another, hopefully less vulnerable, equivalent. This should be possible either on a pre-planned basis, in order to maintain security over time, or as an emergency measure in the event of the mechanism (or some part of it, such as a single cryptographic key) being compromised.[12]
- The security mechanisms employed in the architecture and in implementations based on it should not generally rely on security by ignorance or obfuscation. The publishing of security manuals and other documentation, details algorithms and/or procedural mechanisms should not prejudice the security of the systems protected.

  This assumption is necessary for several reasons:

  - it may be necessary to produce documentation in court to support criminal prosecutions, as is becoming the case in bank ATM system-related fraud accusations (see [1]),
  - it is possible (although this is a speculative comment) that commercial or legal/quasi-legal pressures other than those associated with litigation will require the publication of some security-related information, such as the basis of security policies related to privacy; it may not be possible to assess at this stage the effect on a secrecy-based security system of such future revelations.
  - it is impossible to ensure that all of the procedures will be kept secret, for example in the hands of ex-employees, and the assumption therefore provides greater actual security in the long term, and
  - it is easier to change, for example, cryptographic *keys* than cryptographic *algorithms*, although the latter may become necessary from time to time. Given that the emphasis in designing implementations could usefully be on keeping the required secret aspects of the system as small and as easily changed as possible, it would be better to rely in the architecture on such entities as secret keys than secret algorithms.

[11] Contrast Linn, who believes that it is inappropriate to include authorisation mechanisms in a general framework due to the number of conflicting models in use, and has enshrined that belief in the GSS-API specification ([10, p.483]).

[12] The emergency procedure for mechanism replacement may need to be in the form of recommendations for administrative procedures to select and introduce a new mechanism rather than directly-executable procedures as might be the case for single key replacement.

- A minimum level of security will be required of all entities associated with the network. This is because an insecure entry point to a network can be the starting point for an attack on other parts.

  The security-related requirements set out in the TINA Security Guidelines are regarded by its authors as optional rather than mandatory ([7, p.4]) for TINA-compliant systems. However we suggest that for the reason given above this position should be viewed with caution, and that a minimum level of security defined, below which a system should be regarded as not complying with the architecture.

- Many security breakdowns are due not to attacks on the technical security measures but to failures in the administrative procedures surrounding those technically-based controls ([2]).

  The architecture will need to make explicit those areas for which administrative procedures need to be secure, and ought perhaps to extend beyond the technical solution into specific recommendations for administrative procedures, for which existing notations (such as object-oriented notations) are assumed to be adequate.

- Principals (meaning in this case *people* either operating or using services, or operating and maintaining the network) will be at best almost as bad at generating and remembering passwords as they are at present. Greater general familiarity with computer systems may make them more aware of the requirements of a good password and of the need to keep them secure, and improved passwording mechanisms such as allowing pass-phrases may become available, but the improvement will not be totally reliable and people will not keep to the rules they know they ought to.

  No similar assumptions are made as to the intelligence or otherwise of the human operators of any trusted party system, in keeping with the architecture's view of it as a black box.

- There will be a monitoring activity somewhere, which will check continuously for correct operation and look for anomalies in the systems designed under the architecture. This function will need to be provided with the information it needs to perform its work, and will have to be trusted to the highest level.

  This monitoring function should be independent of the management structures any of the stakeholders if possible, to ensure independence of problem reporting.

- 'We assume that an intruder can interpose a computer in all communication paths, and thus can alter or copy parts of messages, replay messages or emit false material.' ([13, p.993].

- Access controls will be required over service users' use of services and of administrative activities at a finer granularity than the simple user-service connection.

- Implementations of the architecture may be required for legal, political or commercial reasons to be certified based on the architecture to recognised national or international standards (cf. [6, p.6,10]).

  The architecture should not prevent this certification, and may usefully make the process easier by basing some of its design decisions on these standards where they are appropriate and satisfactory.

  The TINA documents assume that by the time networks based on the architecture are deployed, it is likely that the differing sets of standards together with other security evaluation criteria 'will have been merged into a common set of criteria.' [6, p.7]).

- Not all of the entities who need to communicate with each other in an environment protected by any or all of the security systems implemented under the architecture will trust each other. There may need to be provision for communication between arbitrary numbers of non-trusting principals or for trust being achievable by subsets of a nominated group of entities (see [12, p.30]).

Additionally, no assumptions are made concerning low-level computation-based aspects of current security mechanisms, because the implementation details which they represent are outside the scope of the architecture.

Examples of the assumptions which are explicitly excluded as being implementation-related include the un-feasibility of users computing RSA roots or discrete logs of large numbers (see [3, p.258]), and the unforgeability of digital signatures (as assumed in [14]). The intention is that the architecture should remain viable, albeit with implementations modified to use new security mechanisms, in the event of these assumptions becoming unsustainable.

9 The Legal and Regulatory Environment

The architecture will not be able to operate in a vacuum; there will be legal and regulatory requirements to be met, as well as probable code of conduct for any services to be sold to private consumers. These assumptions are described here.

It should be noted that other assumptions about the legal and regulatory basis are made or reflected in other sections of this document.

- Systems designed under the architecture will comply with legal and regulatory requirements regardless of cost. No laws or regulations will be broken to save money.

- for those cases in which different legal and/or regulatory regimes conflict, say in different countries or states, it is assumed that commercial considerations will require that a solution be found; this follows precedents in other spheres of commerce.

  In the absence of long-term commercial predictions for all eventualities, it is assumed that where such conflicts occur a mechanism will be sought which minimises the overall legal liability regardless of all except the greatest cost.

- it is assumed that network carriers, service providers and possibly service users will be required to have a legally-enforceable duty of care to other network carriers, service providers and service users. This will place a burden on these parties which will need to be shown to have been discharged, at least in part by aspects of the implementation of a security architecture.

  As an example, providers of services who place some control over their use may need protection against users and providers of service which have no such protection. This may require additional protection for the network and limited service providers against those who have free direct access to the service, and in order to maintain that free access security measures controlling *but invisible to* the users of those freely-available services may be required.

  Note that questions of law may arise as to whether the duty of care is universal within a multi-carrier network, based on the perception of proximity between the parties involved (see [15, p.43-44])[13]. Whilst such considerations of law are outside the scope of this project, the architecture and its implementations should nevertheless defend its users against possible action by providing a reasonable degree of protection for all network carriers, service providers and service users under the heading of duty of care.

- The final arbitrator of disputes between the stakeholders will be the U.K. courts, or other courts applying an equivalent legal framework of national and international law.

- The provisions and enforcement of contracts between service providers and service users for the use of and payment for services are outside the scope of the architecture. However the architecture will maintain sufficient records to describe unequivocally the events which it controls, and will be able to provide these to support any action between stakeholders.

[13] The same source implies that any action against the regulator of a network for failing to oversee the carrier diligently and thus prevent loss to a service user or provider is likely to fail.

- The courts are likely to be as far behind technological advances in the future as they are now ([2]).

- The courts will seek to base their decisions on parallels with current practice in related business areas, as they have done in the past – hence the mention of the 0898-number model for consumer use at various points in this document.

- Most of the quasi-legal constraints on private consumer use of the services available may be based on either regulations or legally-backed codes of practice rather than legislation. Commercial users of services, and service providers, will generally be less well protected by regulations than private consumers, and may either need or desire a pre-defined arbitration mechanism, or reserve the right to go to law if they deem it necessary.

  Recommendations as to the requirements of contracts between service users and service providers are within the scope of the architecture, as are the need for the architecture to provide evidence of the occurrence and timing of session initiations and completions, but the legal details of contracts are outside the architecture's scope.

- It is assumed that private (as against commercial) service users may not want to have to enter into legally binding, written contracts before using a particular service for the first time. They may be willing to enter into a single contract in order to gain access to the carrier, or to gain access to a group of services. A reasonable model for this might be the current 0898 services, which operate according to these criteria.

  However, the architecture should provide to its users information on how any implicit contracts are established by the use of systems designed under the architecture, and any contractual provisions which are implicit or explicit in the the architecture.

- Commercial service users will be willing to enter into written contracts before they use any service for the first time.

- There will be no legal or regulatory requirement on the carrier to examine the contents of data messages sent between users and services. If such a requirement arises, it will be addressed outside the context of this architecture.

- The accounts of service provider, service user (if corporate) and carrier organisations will be subject to some form of statutory audit or equivalent. Also, formal or informal internal audit functions are likely to exist in any or all of these organisations. The architecture must explicitly provide mechanisms for securing any evidence of transactions required to support such a procedure.

  As stated previously, any record-keeping required after the requirements of the architecture have been met are considered to be outside the boundary of the architecture. As a consequence, the records kept according to the architecture must be complete without reference to any additional information which might be gathered outside it. This may mean that identifications of service users and providers might have to be explicit in the logging records, although possibly encoded.

- The logging and record-keeping required by the architecture to comply with legal and regulatory requirements will be assumed to be permitted by the privacy and data protection legislation and regulations framework in force. Any conflicts between these two sets of requirements and the legal activities required to reconcile them are outside the scope of this architecture.

- No assumptions are made in the architecture as to the legality or otherwise of the use of any particular implementation mechanisms, such as data encryption, used in providing the functionality required by the architecture. It is however assumed that a mechanism will exist which can provide the required functionality and which is available legally to implementers.

- No assumptions are made in the architecture as to the existence or otherwise of particular legal or regulatory requirement to use any particular security-related mechanisms. For instance, the architecture should support designs required to use Clipper, or designs which have another equivalent restriction or no such restrictions at all.
- No assumptions are made in the architecture about the legal status, audit requirements and so on of any trusted party required by an implementation.
- Future legal, regulatory or evidential requirements may mean that systems designed under the architecture will have to be capable of being formally *proven* to be logically free of flaws. The architecture should not prevent such proofs being derived if necessary, although the demonstration of such proofs might not currently be necessary.
- The legal status of digital signatures, both when they are used and in the long term after their algorithms have been broken, is outside the scope of the architecture.
- The question of whether implementations of the architecture will need to support digital cash or other payment systems is outside the scope of the architecture. The architecture is concerned with how a liability is created for the use of services, but not with how that liability is discharged.

  This assumption may be contentious, but is raised to allow the requirement to be aired if necessary.
- Similarly, the consideration of whether any service provider is likely to be considered by its actions to be acting as a bank and thereby involving itself with deposit taking, solvency and other banking regulations, is outside the scope of the architecture.

These assumptions bring to the fore the existence of two types of service users; private users, who are generally protected by regulations and codes of practice, and who are therefore prepared to sacrifice some of the protection offered by a legal contract signed in advance of service use, and commercial service users, who are not so protected and who are more likely to want legally-binding protection on such matters as contractually-specified access to dispute arbitration before they start using a service.

10 Concerning Threats to the System

Assumptions concerning the threats against which systems conforming to the architecture will need to defined themselves as as follows. These are based on the work described in the TINA documents and threat model currently being developed at the Computer Laboratory by Shaw Chuang.

- Threats to systems designed under the architecture may come from inside or outside the set of those authorised to use the system's facilities.
- At some stage the security of the system will have to rely on the integrity of individual people. However, in critical areas division of responsibilities should be a priority, and it should always be possible to identify the person who performed a specific security-related act.
- the threats against systems designed under the architecture are ultimately from *people*. Whilst computer-based agents might perform some of the illicit or illegal activities which are categorised as threats or actions based on threats, the ultimate guiding intelligence will

be human, and security systems should take advantage of any predictable reasons for, and mistakes in, human activities of this kind.[14]

- The currently identified threats against systems derived from the architecture, and against which the architecture will need to defend itself, can be divided in to the following categories (derived from [6, p.8-9]):

  – loss of accountability for actions, whether by impersonation of a valid system user (of any kind) by another person not so authorised, or the repudiation of an act by an authorised person,
  – loss of confidentiality of data or identity,
  – loss of integrity of data held by service providers, network carriers and/or the security-related systems, and
  – loss of availability of network or service.

  It may be assumed for implementation purposes under the architecture that no threats will arise which cannot be included in one of the above categories. However the architecture itself should be sufficiently flexible to allow for other threats to be countered as far as possible.

- It will be possible to prioritise the different types of threat in respect of a particular service. This prioritisation may differ from service to service; for example, occasional non-availability of a home banking system may be deemed to be less important than the occasional corruption of transactions, whereas the provider of a downloaded entertainment system may require high availability, if necessary at the cost of the occasional failed transaction. It will therefore not be possible to produce a prioritised list of threats which covers equally well all services (or parts of services with different security requirements) protected by an implementation of the architecture.

  This means that the architecture will need to support different prioritised threat models for different services simultaneously, which in turn supports the need for different security mechanisms to be present and available simultaneously.

- Some or all of the threats identified will be considered in the context of the cost of providing differing levels of protection against them. This assumption does not apply to the costs of meeting legal and regulatory requirements.

11 Interactions with the Architecture

In order to make use of the facilities offered by systems developed under the architecture, principals will need to interact with that system in some way. Assumptions related to this aspect of the architecture are described below.

- The architecture will make no assumptions concerning any aspect of service users, service providers and network carriers, and of trusted parties if any, except as specified elsewhere in this document.

- In determining security mechanisms, the architecture could be considered to be either

  asymmetric i.e. services and users are assumed to be different types of thing), which begs many questions as to the nature of 'service providers' and 'users', or

[14] This may, for instance, result in an examination of how previous security failures were detected, and the motives and errors of the perpetrators. Information kept confidential for security or commercial reasons may be obtainable by the security architecture development team for this purpose, and this information should be examined where possible.

symmetric , i.e. services and users are not to be distinguished and are to be provided with the same sets of abilities and rights (and obligations.

In order to preserve generality, it is assumed that the requirements of service providers and of users are not necessarily the same, and that an asymmetric model be adopted. Such a model can support instances of interactions which happen to be symmetric. However, in making this assumption no position is being taken as to the relative importance of service providers and service users.

- an entity interacting with systems designed to meet the architecture may act as more than one type of stakeholder *at different times* – a service user might be a service provider. It is assumed that in its interactions with the system the entity will not act as more than one stakeholder at the same time, the stakeholder role being acted under will be identifiable at all times and any change of stakeholder status will be made explicitly and be visible to all entities in the system.

  Query: who changes the status? is it the stakeholder, or is it the security managers; is it done automatically at times?

- The nature of the data messages to be sent through the network, with regard to, for example, type of information, content and service quality requirements, is outside the scope of the architecture. The architecture should make no assumptions about the nature of the data traffic – frequency, bandwidth, burstiness, etc.

- The nature and ordering of the interactions expressed in the data messages between service providers and service users via the network cannot be determined in advance.

  The architecture should make no assumptions about the nature of the user/service protocols used once the architecture's requirements have been met.

- Any or all of the services to be secured using the architecture may require payment to be made for their use, or have usage quotas on their use. Conversely, some services might not be subject to any controls over access or use, and the architecture should allow an application to be changed from being chargeable to free access and *vice versa* by modification to its data structures rather than by having to recode the applications.[15]

  Note in this context [6, p.14]: 'Appropriate audit mechanisms should be used to map each *chargeable* action to a specific customer' [our emphasis]. This reflects the concerns of [6] and [7] with commercial considerations.

- For any service for which a charge is made, charges may be levied for the use of the network, on a connect time/distance, per unit of use or any other basis. Payment may be required in advance or in arrears.

  The details of charging mechanisms for service use and associated record-keeping are outside the scope of the architecture. The architecture will however need to ensure that sufficient information is kept in logs or the equivalent to confirm that connections were made and broken for specific users at specific times. Additionally, any assumptions regarding the traceability of activities may become problematical in the context of potential requirements for privacy and anonymity.

  The architecture may also need to provide some messaging mechanism from both service provider and user to the network carrier to notify of session connection and disconnection; this would act both as a safeguard against continued charging for apparently finished time-based connections (cf. international telephone calls), and as a mechanism to enable a carrier to reduce the chance of erroneous or malicious picking up of a disconnected connection by an unauthorised party who would thereby inherit that connections' rights and thus circumvent

[15] This is an example of an instance in which the architecture should recommend strict administrative controls to prevent illicit use of an option which makes a service free to use.

the controls in the architecture. Although this event should be allowed for in the systems developed under the architecture, additional safeguards can do no harm and might prevent problems arising in the event of an error on the architecture's implementation.

- It cannot be assumed that the security domains of service users, service providers and network carriers will overlap at any point. In order to maintain generality these three must be seen for architecture purposes as being separate domains.

  Note that this assumption conflicts with that of [6, p.5],[16] in which the service and user domains are seen as being sub-domains of the TINA architecture as a whole.[17] This implies a considerable degree of control over user and service provider activities and system design decisions.

- A number of service users and/or service providers will be identified over time whom it will be desired to exclude from access to the security envelope provided by systems implemented under the architecture. The reasons for which this might arise are a matter of security policy and outside the scope of the architecture; these might for instance include delinquent behaviour or failures to pay bills. These excluded parties may be human or corporate.

  Additionally, any computer-based entities representing these excluded parties should be excluded at the same time as the human or corporate party concerned.

  Note that Needham and Schroeder explicitly did not consider the problems of '...trying to prevent communication between particular principals in order to enforce restrictions on information flow.' ([13, p.993].

12 How the World is Organised

No explicit or implicit assumptions are to be made in the architecture as to how the world within which the architecture will operate is organised beyond accepting the existence of service providers, service users, network carriers and (possibly) parties trusted by any or all of these three. Each of the matters discussed in this section is outside the scope of the architecture.

However it is important to describe the possible areas and types of organisations in order to ensure that the architecture will be able to work within any likely set of values. Essentially the architecture must be sufficiently flexible to meet any currently envisaged organisation.

- The organisational structure (if any) within each of the service provider, service user, network carrier and (if any) the trusted party,[18], is outside the scope of the architecture.

  Note that private consumers as well as corporate bodies will need to be accommodated in this category. In terms of the unit connected to the network and identifiable as a single 'service user' for billing or identification purposes, private service users could constitute a hierarchy, e.g. a family with parents and children, or a flat structure, such as might exists in the case of a shared dwelling with one shared network access point.

  Any assumption made in this area will affect whether any delegation performed in an organisation is hierarchical or peer-to-peer. Also the trust relationships within the organisation might well mirror the organisational structure since both relate to the power structure within that organisation.

- The naming structure used by the sender of a message via the network to identify and gain access to the intended recipient is not to be made explicit in the architecture.

[16] However, the possibility of these being in different *administrative* domains is accepted – see [6, p.5-6].

[17] 'The TINA services and their customers reside in a domain that is under the control of a TINA network operator.'

[18] For example are they organised in a hierarchical, flat or other manner?

Current models for this include hierarchical computer network naming systems, and these parallel familiar models from the real world, such as building addresses and telephone numbers – although in the latter case some of the structure is implicit to the end user.[19]

The architecture should not be concerned with how the 'other end' for any transmission is identified, only that this can be done and that the information can be sufficiently well-defined to allow messages to reach the intended party and ensure that any identifications made in logs and audit trails are meaningful.

- It will not be possible in all cases to restrict the protection offered by implementations of the architecture to members of closed groups defined by pre-use agreements, as is the case for current EDI and equivalent systems whose legal positions are defined by framework agreements. There will be not necessarily always be a contract to which reference can be made in considering difficult legal areas such as the validity of digital signatures.

  Adopting this assumption prevents reliance on predefined agreements to cover such aspects as definitions of digital signatures. There are as yet (April 1995) no legal precedents concerning the status of digital signatures in such an 'open EDI' environment.

13 Concerning Providers of Services

There is a number of specific assumptions to be made with respect to the providers of services (content providers, service retailers and service brokers) under the architecture. These assumptions are described here.

- Service providers will vary in size and sophistication from small organisations with a service to sell but without any understanding of the security implications of their actions, to large companies equivalent to current database publishers with a good understanding of security and how to achieve it. The systems developed under the architecture must accommodate all types of users including both extremes.

- The question of determining whether service providers have legal title to the information and/or programs comprising any services which they wish to protect using an implementation of the architecture, and of how they are to demonstrate this before they are allowed to take advantage of these security services, is outside the scope of the architecture.

- It may be necessary at times (not necessarily discernable in advance) to determine the source of information provided through implementations of the architecture, for instance to ensure that copyright and other intellectual property rights are being respected.

  This may require the architecture to support the provision and checking of such measures as relevant auditable logs of access to services in general terms (start time, end time). Data or program protection mechanisms such as the retention of specific information to determine the exact data used or received, or support for proof of ownership via the watermarking of software, are outside the scope of the architecture. It is assumed that s measures can be implemented without conflicting with privacy and/or anonymity requirements.

  Note that Muftic differentiates between illegal copying and illegal distribution of software ([12, p.64, reference GOLD87]); we may need to make the same distinction in the context of information (not necessarily software) available over a network..

  currently in scope of our architecture – should it be?

- Service providers will have available a general-purpose computing resource which can be used to perform some of the work required by the architecture.

[19] Note that networks can be organised non-hierarchically – consider railway station names.

- There will be a number of existing services which service providers will wish to provide through the media supported by the architecture. The architecture will need to support the conversion of these existing applications to work under it with a minimum of effort.

- Different services, and different parts of single services, provided through the same carrier may require different levels of, and different mechanisms to support, security (cf. [6, p.10]).

- The same service provided by the same provider via the same carrier may require different levels of, and different mechanisms for, security for different users.

- The same service provided via the same carrier may require different levels of, and different mechanisms for, security for different facilities offered within what is perceived as being the same 'service' by service users and/or network carriers.

  The details of any subdivision of services made for this reason, and the mechanisms to secure such subdivisions, are outside the scope of the architecture. However the architecture will give service providers sufficient valid information concerning, for example, the identity of the service user, to allow the service provider to implement such systems. The decision as to the information needed by the service provider, and therefore what is to be available from the architecture, will be made during the research leading to the design of the architecture.

  Note in this context the requirement set by [6, p.15]: 'Include any security requirements specified by the customer in the service level agreement' However it is not stated here whether the 'customer' is the service provider, the service user or the network carrier.

- A service provider may require the use of security mechanisms over his or her services which are outside the architecture in terms of functionality, fineness of grain, quality of protection etc. Systems designed under the architecture should not prevent this, and should support it, for example by means of optional fields within dialogues (as does Kerberos).

- The same service provided through different carriers may require the same security-related information to be available. For example, a service user may require the ability to access the same service through different carriers without having either to present different information or devices to gain access or to use a different mechanism.

  The architecture should not prevent this possibility.

- A service provider may require changes to the levels of, and mechanisms for, security at any time for any service on any or all networks.

- A provider of a service should be able to delegate (effectively subcontract) their rights and obligations to provide a service to another service provider (who may be an existing service provider, an existing carrier user, or neither), with the minimum of modification to the information held in the architecture's data repository.[20] The architecture must provide mechanisms to support securely, and provide evidence of, such delegations.

14 Concerning Users of Services

There are some specific assumptions which need to be made concerning the customers for and users of services under the architecture. These are listed here.

- Service users will comprise both commercial and private users. The latter will essentially form a consumer market for services provided by systems protected by measures based on the architecture, and will generally be comparatively naive about security requirements, procedures and implications within computer-based networks.

[20]The possible need to inform or agree such a change with the network carrier, and the formalisation of agreements between delegator, delegatee and network carrier are outside the scope of the architecture.

- Service users will generally not have available a general-purpose computing resource which can be used to perform some of the work required by the architecture. Specific devices, as noted below, may be available.

  Note that

  – Needham and Schroeder assume that 'Each principal has a secure environment in which to compute, such as is provided by a personal computer or would be by a secure shared operating system.' ([13, p.993]. Our assumption here is that such a facility will be available to all service users, but may be seen by any or all of those users as a dedicated 'key' to obtain access to one or more services rather than a computing resource of any kind.

  – different sets of assumptions regarding the accessibility of general-purpose computing facilities to service users, service providers and/or network carriers may result in a requirement for, or a limitation on the ability to support, different authentication and authorisation mechanisms.

- Many potential users of the service will already have electronic devices, supplied and/used for purposes unrelated to those supported by the architecture, which can be used to implement security features under the architecture. These devices might for instance be smart cards of some sort.

  This means that the managers of systems implemented under the architecture will be able to rent space on another issuing body's card should they so wish.[21] It also implies that such service users will be at least generally familiar with the principles of use and management (such as keeping PINs secure) of such devices.

- Service users will be willing to keep and use different devices or information to gain access to different services, as long as the number of such devices or sets of information is limited to a reasonable figure for most service users.

  In the case of smart cards or similar devices, users will (probably) be prepared to have to obtain and use another card to obtain new or additional services. This may seem to conflict with the assumption that they will not want to sign contracts for individual services, but the suggestion is that users will be prepared to use a general card for a number of specific services as against a card for each service they wish to use beyond a small number.

- The above leads to the further assumption that a smart card issuer/manager might not necessarily be either a service provider or a network carrier. The card manager might be an example of the 'trusted party' concept.

- No assumption will be made in the architecture as to the specific functionality to be available from or provided by devices such as smart cards. Such functionality might include such facilities as the calculation of nonce keys, encryption and/or decryption (see, for example, [8]), or the identification of the valid holder of the device by means of PINs etc. The mechanisms, provision and use of these facilities is an implementation aspect outside the scope of the architecture.

- It is assumed that those potential service users who do not already have any identification or other device needed to use services will be prepared to obtain one in order to use a service which demands its use.

[21] This has commercial disadvantages – we could be bumped by another user paying more, and security disadvantages – we would only be as secure as the card issuer, and would be unable to secure our services against the card issuer itself. However, the architecture should not prevent to adoption of such a mechanism, i.e. such considerations should be outside the scope of the architecture.

- Service users will be assumed to want to gain access to and use services in as convenient a manner as possible. If the mechanisms are complex, users will tend to try to circumvent them. Similar remarks apply to those who operate services and maintain the network. All security mechanisms should be both as simple as possible to use, and as un-circumventable as possible, in terms of both security and procedural short-cuts.

- As a specific case of the desire of users to gain convenient access to a service, in order to identify him- or herself to a service the service user might want a choice of either some physical device[22], or another mechanism which allows transparent and concurrent sharing of the right to a service.

- A user of a service should be able to delegate their rights to use a service to another (who may be an existing service user, an existing network user, or neither), preferably without the need to inform the service provider, with either the delegator or the delegatee paying the bill at the delegator's decision.[23]

  The architecture must provide mechanisms to support securely, and provide evidence of, such delegations.

- Service users will need to be able to pay for their service use by any acceptable means, including pre-payment, billing and post-payment, and charges against an account at a third party. Additionally, the carrier may be required to collect payments from service users and redistribute them to service providers. As noted above, access controls and charges should not be assumed to apply to all services and/or users.

- Service users may in certain reasonable circumstances wish to act anonymously; an example is the use of a confidential help line. This anonymity is currently under threat from legal, regulatory and private bodies.

  It is assumed that anonymity will be allowed to continue under future regulatory and legal regimes, and will be a required facility of the architecture.

  [6, p.8] notes 'loss of anonymity' as a possible risk if confidentiality is lost. This assumes that the legal and regulatory framework will allow such anonymity in practice.

  One mechanism to enable the easy use of anonymous help lines is to implement them as free, no-authentication systems, in which case no personal data would need to be kept either for billing or for authentication purposes. However, this would conflict with another assumption that no assumptions are to be made concerning commercial matters, and would still require some control over the activities of such anonymous users under the duty of care to other contracted parties.

- Some service users will need to be assured that the confidentiality of their actions (which service was used, and the use made of that service) can be maintained where they deem it necessary for commercial or personal privacy reasons.[24]

  This may be problematical if national security services are allowed to monitor activities of commercial users of services, and those services include industrial espionage as part of their remit. The possibility of interception of the transmission of, say, the design parameters for a new microprocessor being sent to a chip foundry, could have considerable implications for commercial use of the services.

[22] which might be considered more secure, and, if not physically copied, shows that the device at least is present for a service to be used, but which can only be transferred from one user to another and could not be shared transparently and simultaneously among several users

[23] The service provider might reasonably demand in the latter case that the delegator guarantee that the bills will be payed, since the provider will not have prior knowledge of the credit status of the delegatee. In this case, legal recommendations as to contract terms might form a part of the architecture.

[24] Muftic notes the use of digital pseudonyms, which might implement this requirement([12, p.31]).

- The distinction between human and computer-based service users when using services is not always useful now, and may become less so in future. Consider the difference to a World Wide Web server between a remote human terminal operator and an artificial intelligence-based data capture program. Additionally, automatically operating programs might usefully be employed at times to take advantage of cheap service charges at certain times of day, such as at night. These reasons may indicate that facilities will be required to allow computer-based 'users' to gain access to services without human intervention.

  It may also be necessary to assume that such computer-based systems will never have as great a level of security clearance as humans. However to support this assumption it will need to be possible to differentiate between people and computer-based systems.

15 Concerning Security System Providers

The assumptions concerning those responsible for providing this security architecture and implementations of it have been deliberately minimised to ease the task of considering the environment within which the security system is to operate.

These assumptions are:

- the providers of security systems are not necessarily trusted by the other stakeholders.
- security systems implementing the architecture can be certified or otherwise evaluated to provide a sufficient degree of comfort for its users to use them.[25]
- the commercial risk of using the implementations of this architecture will be deemed acceptable by those who are responsible for commercial risk assessment; note that this is a different way of achieving the same result as the immediately preceding assumption.
- mechanisms outside the scope of this architecture will be in place to assure its users that the legal risk associated with the use of implementations of the architecture will be acceptably low; for instance, there will be a perception, however engendered, that the security system will falsely accuse its users of fraud and provide faked 'evidence'

16 Concerning Network Carriers

There are a number of assumptions which need to be made regarding the organisations providing the equipment and services comprising the network used by service providers and service users to communicate. These are listed below.

It is worthwhile reiterating here the assumption stated earlier in this document that the control of service providers' and users' initial access to the network will be maintained outside the architecture.

- It cannot be assumed that all messages sent in managing networks, say to monitor or control the activities of a network switch, will be directly originated by a human. It is possible that, now or in the future, some of these commands will be generated automatically. However, it is assumed that some human-based authority, based on an identifiable individual acting in an identifiable role, will be required to perform any network management function.

[25]The certification or audit mechanisms employed to provide this comfort are outside the scope of this architecture

- There will be an arbitrary number of network carrier organisations involved in any link between a service provider and a service user. These carriers will not necessarily have the same legal, quasi-legal or policy-derived security concerns and policies.

  It may be that for some modelling purposes the link between service provider and user be seen as a single link (say for end-to-end authentication protocols), but this will not influence the more general assumption stated above.

- Network carrier organisations will generally have a good understanding of security with respect to networks. It may be more dangerous to assume anything about their degree of knowledge about service users' and providers' security requirements.

- Network carrier organisations will have available a general-purpose computing resource which can be used to perform some of the work required by the architecture.

- Any existing controls over service providers' and users' access to the network will not of itself necessarily provide a suitable mechanism to secure the demands of the architecture for user/service security. Additional measures may be needed under some (but not necessarily all) circumstances.

- It is assumed that more than one route, involving an arbitrary number of network carriers, will be available to connect a particular service user to a particular service provider. Each of these network carriers may differ in any respect other than the minimum necessary technical agreement to achieve sufficient communication to support the architecture.

- Different network administration function, and different parts of the same function, may require different levels of, and different mechanisms to support, security (cf. [6, p.10]).

- The same network administration function provided by the same carrier may require different levels of, and different mechanisms for, security for different administrators.

  This may be especially true for cases in which delegation to administrators belonging to outside carriers occurs.

- It is assumed that the architecture may be required to support different security policies in different political, economic or geographic sub-domains of the network.

  [7, p.20] notes that security policies might differ in different domains which require to communicate.

- Different data elements maintained by the network carrier may require different levels of, and different mechanisms to ensure, security (cf. [6, p.10]).

- Network carriers may at times require to delegate any aspect of the work of network management, between authorised individuals within an organisation, between organisations, and across security and/or administrative domains.

  Failure to include this assumption may prejudice the network management process, especially when major network problems arise and co-operation is needed between the administrators in different security domains in order to solve them.

17 The TINA Documents – Summary of Differences

British Telecom have produced two documents relevant to and/or extending the assumptions given above ([6] and [7]).

The general attitude underlying the work described in these documents can be related to the comment that [6] '...concentrates in particular on the stakeholders with the most to lose – the service and network providers' ([6, p.4]). The rationale for having a security architecture is strongly biased towards protecting the revenues of those who charge for services; see [7, p.3]. Additionally, their methodology is based on first determining the requirements of a security system; neither document makes use of any possible prior investigation of the underlying assumptions such as that presented here.

During the work presented here, we have noted differences between the assumptions which we have made and those explicit and implicit in the TINA documents [6] and [7]. These can be summarised as being due to

- our taking a wider definition of the stakeholders in the security architecture; the TINA documents by contrast limit the stakeholders in the architecture to those commercially involved,
- our ignoring the wider commercial interests of any of the stakeholders, other than in our assumption that the entities involved in the operation of the architecture will not necessarily trust each other; however the safeguarding of short-term commercial interests such as protection of any claimed intellectual property rights and the need to protect chargeable resources against fraudulent use are explicitly part of our architecture's remit,
- our giving legal (and possibly ethical) considerations a higher priority than is possible for a purely commercial enterprise,
- our desire not to prioritise the commercial rights of the different stakeholders in general terms; the attitude displayed in the TINA documents is oriented towards protecting the network carrier and service providers,
- our desire to make more explicit the assumptions concerning when security will deliberately be breached (for such as national security reasons),
- our belief that in particular small service providers may not be aware of security implications,
- our rejection of the single security domain including both service providers and users under TINA control explicit in the TINA documents; this affects both the attitude to the service providers and users and the independent place of security audit and monitoring functions,
- our assumption that in the long term services will need to be available, at least to domestic customers, without the need for explicit, individually signed, written contracts to be completed, and that limitations on liability and so on which would have been set out in such contracts may not be available, and
- our desire to distance the architecture from any particular assumptions regarding design or implementation, such as whether a trusted third party is necessary for the architecture or the bandwidth limitations of transmission lines.

To summarise, we regard the architecture as being a long-lived artifact, and therefore assume that it must remain independent of any specific technical, security or commercial assumptions or interests.

18 Assumptions Still to be Considered

There are some areas in which the decision as to whether an assumption is needed, or as to which of more than one possible value for an assumption is to be included, is yet to be concluded. These matters include:

- how the lack of trust between parties will affect other assumptions, particularly regarding those parties and how they interact,

- Any assumptions regarding the trust structure within the architecture. Note that the trust structure is assumed not necessarily to be identical to the organisational structure, which I have excluded as far as possible; this seems reasonable when, for instance the position is considered of internal audit functions which report to the highest authority in an organisation, are highly trusted but which have no 'power' in direct organisational terms.

  As an example of trust structure, many authentication protocols assume that trust is hierarchical – see Lampson ([9]) for an explicit example of this, whereas PGP assumes that all users and servers are equal and are able to manage their own trust relationships. Other systems, such as PGP, assume a peer-to-peer mechanism.

  It may well be that any reasonable trust structure can (or should) be capable of being accommodated within the architecture, which would accord with the attempts made previously in this document to ignore the internal structure of service providers, users and network carriers.

References

[1] R.J. Anderson. Liability and computer security: Nine principles. In D. Gollmann, editor, *Computer Security – ESORICS '94*, number 875 in Lecture Notes in Computer Science, pages 231–245. Springer-Verlag, 1994.

[2] R.J. Anderson. Why cryptosystems fail. *Comm. ACM*, 37(11):32–40, 1994.

[3] D. Chaum and E. van Heyst. Group signatures. *Lecture Notes in Computer Science*, 547:257–265, 1991.

[4] D.E. Denning. Secure personal computing in an insecure network. *Comm. ACM*, 22(8):476–482, 1979.

[5] A. Herbert. An ANSA overview. *IEEE Network*, 8(1):18–23, 1994.

[6] S.D. Hubbard. TINA: Internal team task 1 core security requirements for TINA. Technical report, British Telecom, Nov 1993. Version 1.0.

[7] S.D. Hubbard and M.A.Z. Rejman-Greene. TINA: Security guidelines for TINA. Technical report, British Telecom, Mar 1994. Version 1.0.

[8] M. Krajewski, J.C. Chipchak, Chodorow D.A., and Trostle J.T. Applicability of smart cards to network user authentication. *Computing Systems*, 7(1):75–89, 1994.

[9] B.W. Lampson. Authentication in distributed systems. In S. Mullender, editor, *Distributed Systems*, pages 543–580. ACM Press/Addison-Wesley, 1993.

[10] J. Linn. Generic interface to security services. *Computer Communications*, 17(7):483–491, 1994.

[11] D.R. McAuley. *Protocol Design for High Speed Networks.* PhD thesis, Computer Laboratory, University of Cambridge, 1990. Available as Technical Report No.186, Computer Laboratory, University of Cambridge.

[12] S. Muftic. *Security Mechanisms for Computer Networks.* Ellis Horwood, 1989.

[13] R.M. Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Comm. ACM*, 21(12):993–998, 1978.

[14] B. Pfitzmann and M. Waidner. Unconditional byzantine agreement for any number of faulty processors. *Lecture Notes in Computer Science*, 577:339–350, 1992.

[15] W.V.H. Rogers. *The Law of Tort*. Sweet and Maxwell, 1989.

[16] J. Rumbaugh et al. *Object-Oriented Modeling and Design*. Prentice-Hall International, Inc., 1991.

[17] V.L. Voydock and S.T. Kent. Security in high-level network protocols. *IEEE Communications Magazine*, 23(7):12–24, 1985.