# REGIONAL BLACKOUTS: PROTECTION OF BROADCAST CONTENT ON 3G NETWORKS

**Alexander W. Dent, Allan Tomlinson**

Mobile VCE Research Group, Information Security Group,
Royal Holloway, University of London, Egham, Surrey, TW20 0EX, England
alex@fermat.ma.rhul.ac.uk, allan.tomlinson@rhul.ac.uk

## Abstract

One of the driving forces behind the development of 3G systems is the potential to deliver complex content to consumers. This is evident from the growing collaboration between broadcast and mobile network operators, and the expectation that future broadcast receivers will be able to forward content to mobile devices. One challenge in providing such a service is the requirement for content protection. An aspect of this that is particularly relevant to mobile systems is the ability to control where content is viewed. Although 3G networks can provide location of a user's receiver, this device may be in a different location from the device that renders the content. Thus the provider cannot be certain where the content will be viewed. This paper proposes two protocols that will provide the location of the end device in a secure manner that can be trusted by the content provider.

## 1 Introduction

Advances in mobile communications technology have provided the potential to deliver many new services to subscribers. New businesses models are anticipated where interactive multimedia services will be available to mobile subscribers instantly and at any location. The removal of barriers to the delivery of such services raises issues of content protection and digital rights management (DRM).

The nature of digital multimedia content lends itself to theft by copying. Content providers are naturally concerned about this, and are seeking the development of technology to minimise this risk to their business [9]. Consequently there is increasing interest in DRM languages [1, 8], and system architectures [3, 12, 13].

Successful DRM architectures must be trusted by the content provider. This requires trust in the platform that supports the DRM system, and trust in the sources of data that determine the conditions under which the content may be used. The ability to manipulate these usage conditions represents a threat to any DRM system.

An example of this threat is in the implementation of regional blackouts. Broadcasters are often required to restrict broadcasts of certain content to specific geographical regions, or specific dates and times. The reasons for doing so could be to meet local regulations, or to meet commercial terms agreed with the content provider.

For example, coverage of a sporting event may be forbidden in regions close to the stadium while the event is taking place. Subscribers in other regions may pay to view the live event. Immediately after the event the content may be freely available in all regions.

Currently, broadcasters use conditional access systems to scramble such services in a manner that can only be unscrambled by receivers with specific embedded regional codes. The assumption is that the receivers remain relatively static. Mechanisms have been proposed to track the location of receivers using telecommunications return channels, GPS technology, and satellite ranging [6]. These solutions however, are not intended to apply to low cost mobile devices with no satellite receiver.

The problem is compounded where receivers have the ability to store and forward content. Current solutions ensure that the *point of reception* is outside the blackout region. What is now required is proof that the *end user* is outside the blackout region. To achieve this, the location of any device capable of rendering content must be made available to the DRM application in a trustworthy manner. Moreover, it is the user who is most likely to try to deceive the application.

Other authors have focussed on problem of location awareness to assist with routing [7] or to determine a device's location with respect to some fixed transmitter [2,10]. The issue of trust, however, is not relevant to these applications. The recently proposed Echo protocol [11] does address the problem of trustworthy location data, but this mechanism is designed for physical access control and uses ultrasonics and fixed transmitters. It is not intended to cover large geographical regions.

This paper examines how secure time and location information can be provided in a manner that can be trusted by a third party. The use of trusted hardware is considered followed by the proposal of two protocols that address the problem without using trusted hardware. The methods proposed also reduce the opportunity to forward legitimately received content on to third parties.

## 2 Preliminaries

The model under consideration is illustrated in Fig. 1, and is designed to reflect the following scenario: A broadcaster purchases content from a content provider in order to sell this content to a subscriber. The content provider is free to place restrictions on the location and time that his content may be viewed. The broadcaster then delivers the content to the subscriber's set-top box. The set-top box is then able to forward the content, via an intermediary network, to the user's laptop, where it can be rendered. There are five entities in this model:

A **content provider** who provides encrypted digital content.

A **broadcast network** that initially delivers the content to the user.

A set top box, (STB) which is the **initial receiver** of the content. This device is controlled by the broadcaster and provides a secure platform to manage viewing rights. The user, however, may be able to manipulate data that enters and leaves this device.

An **intermediary network** that forwards the data from the set-top box to the end viewing device. Note that part of this network could be under the control of the user.

An **end device**, that is under the control of the user, on which the content is to be finally displayed.

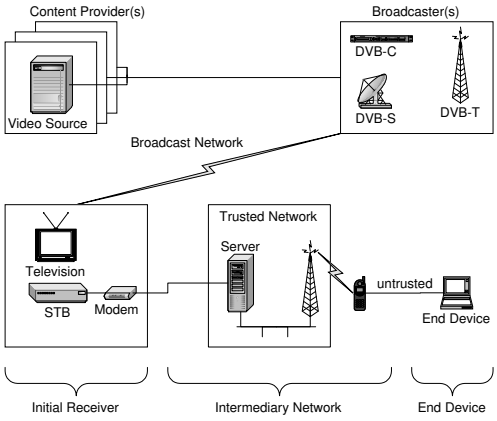


Figure 1: The Network Model

An important aspect of the above model is the intermediary network. The set-top box and the end device do not communicate directly but are connected by an intermediary network. It is important that some part of this intermediary network is trusted by the content provider. However, *part* of the network may be controlled by the user, this models the situation where the end device is connected to a trusted network via a second device controlled by the user such as a cellular phone.

This means that the user may attempt to alter, delete or insert messages at any stage between the set-top box and the end device. This also allows the user to forward the content some distance away from the trusted network.

| | |
|---|---|
| $STB$ | denotes the initial receiver |
| $ED$ | denotes the end device |
| $IN$ | denotes the intermediary network |
| $LS$ | denotes the location server closest to $ED$ |
| $CA$ | denotes a trusted certification authority |
| $\text{Cert}_X$ | is a public key certificate for entity $X$ |
| $K_{X,Y}$ | denotes a secret key possessed only by $X$ and $Y$ |
| $ID_X$ | denotes the unique ID of entity $X$ |
| $R_X$ | is a random number issued by entity $X$ |
| $\text{t}_i$ | is a time stamp issued at time $= i$ |
| $\text{dt}_{i,j}$ | is the time interval between $\text{t}_i$ and $\text{t}_j$ |
| $\text{dt}_{max}$ | is an upper limit on a time interval used to determine how close, geographically, one device is to another. |
| time | is the time and date data provided by $LS$ |
| loc | is the location data provided by $LS$ |
| $E_K(Z)$ | is the result of the encipherment of data $Z$ with a symmetric algorithm using the key $K$ |
| $MAC_K(Z)$ | is the Message Authentication Code, generated by hashing data $Z$ with the key $K$ |
| $S_X(Z)$ | is entity $X$'s private signature transformation operating on data Z |
| $V_X(S_X, Z)$ | is entity $X$'s public verification transformation operating on $X$'s signature $S_X(Z)$, and data Z |

Table 1: Notation

In describing the protocols the notation listed in Table 1 is used, and the following conditions are assumed:

1. $STB$ and $ED$ have a secure execution environment.
2. $STB$ and $ED$ have a tamper-proof data storage area.
3. All cryptographic processing on $STB$ and $ED$ is carried out in the secure execution environment.
4. Only applications running in the secure execution environment have access to the tamper-proof data storage areas in the $STB$ and the $ED$.
5. At least one authenticated key, $K_{S,E}$, is shared by $STB$ and $ED$ and stored in the tamper-proof storage.
6. At least one of the $ED$ or $STB$ possesses a public verification transform, $V_{CA}$, for a certification authority $CA$, stored in its tamper-proof data storage area.
7. The initial receiver, $STB$, possesses a DRM application stored in its tamper-proof data storage area.
8. The initial receiver, $STB$, has knowledge of the usage criteria for each service received.

## 3 Using Trusted Hardware

An obvious way to provide trustworthy location data involves using hardware that can be trusted by the content provider. The end device may have a trusted hardware component that provides its current location and the current time. A global satellite navigation system such as GPS [5] or the proposed Galileo system [4] could accomplish this. This solution, however, is likely to be expensive.

Alternatively, the end device could have *direct* access to the trusted part of the intermediary network. The network could then offer an extra service: on request, the network could report the current time and the location of the gateway from which the end device is receiving content.

If the content provider could trust that the DRM application is receiving information directly from the third party network then this would solve the problem. This solution, however, restricts the end device to those that can be connected directly to the trusted network. For example, it would be possible to view the content on a cellular phone, but it would not be possible to forward the content to a laptop connected to this phone.

## 4 A Software Based Protocol

If the link between the trusted network and the end device cannot be trusted then there is a fundamental problem: although it is easy to ensure that time/location data has come from a trusted network, it is difficult to ensure that the data has not traveled a long distance. We say that data that has not been sent too far is *near*. It is also important to know that data is not a replay of some earlier execution of the protocol. We say that data that has been recently generated (in particular data that is not being replayed) is *fresh*. It is necessary that the data can meet both these conditions if it is to be trusted.

Consider the following service offered by the intermediary network. Suppose that, on receiving a random nonce from an end device, a network gateway signs a data string consisting of that nonce, the gateways location and the current time. Obviously the use of digital signatures implies the need for the end device to trust the network's public key, but this could be solved by means of a certificate supplied by the content provider. An end device could then accurately validate its time and location by sending a nonce to the nearest trusted network gateway and checking that: the response has been signed correctly (entity authentication); the response includes the correct nonce (freshness); and the time taken between sending the request and receiving the response is less than some threshold determined by the content provider (nearness). If all of the above conditions hold then the DRM application can trust the time and location information contained in the response. The following describes such a protocol in detail.

In addition to the conditions of section 2 it is assumed that the end device, $ED$, possesses a DRM application stored in its tamper-proof data storage area. The protocol is initiated by the user requesting a service on $ED$ which causes the DRM application to be loaded and the following steps executed. The protocol is illustrated in figure 2.

1. $ED \rightarrow STB$ :
   Request for usage criteria || $ID_{Service}$
2. $STB \rightarrow ED$ :
   $ID_{Service}$ || usage criteria || $MAC_{K_{S,E}}$(usage criteria)
3. $ED$ calculates: $MAC_{K_{S,E}}$(usage criteria)
   and compares the result with the received MAC to verify the origin and integrity of the usage criteria.
4. $ED \rightarrow IN$ : Request $ID$ of nearest location server
5. $IN \rightarrow ED$ : $ID_{LS}$
6. $ED \rightarrow CA$ :
   Request certificate for location server || $ID_{LS}$
7. $CA \rightarrow ED$ : $\text{Cert}_{LS}$
8. $ED$ executes $V_{CA}(\text{Cert}_{LS})$ to verify $V_{LS}$
   If $V_{LS}$ is verified, then it is stored in the tamper-proof data storage area on $ED$.
9. $ED$ generates a random nonce, $R_{ED}$
10. The DRM application running on $ED$ generates $\text{t}_i$ and stores it in the tamper-proof data storage area.
11. $ED \rightarrow LS$ : $R_{ED}$
12. $LS \rightarrow ED$ : time || loc || $S_{LS}(R_{ED}$ || time || loc)
13. The DRM application running on $ED$ generates $\text{t}_j$ and compares it with $\text{t}_i$. If $\text{dt}_{i,j} > \text{dt}_{max}$ then $ED$ is geographically too far from $LS$ to provide reliable data. Otherwise, $ED$ checks the validity of the signature provided by the location server using $V_{LS}$.
    This verifies the origin of the time and location data and verifies that the data has not been replayed. The DRM application can then check the usage criteria and request, or halt, delivery of the service.
14. $ED \rightarrow STB$ :
    Request for service || time || $MAC_{K_{S,E}}$(time)
    The MAC authenticates the origin and integrity of the request and prevents replay.
    This is necessary to defeat a user who is receiving a free service from injecting a request for a restricted service in step 1, blocking subsequent messages, then injecting or replaying a request for service at this step.
15. $STB \rightarrow ED$ : $E_{K_{S,E}}(Service)$
    $STB$ then delivers the service identified by $ID_{Service}$ received in the preceding request for usage criteria.

16. The protocol repeats from step 9 to step 13 at regular time intervals determined by the DRM application. This ensures that ED remains within the permitted location.

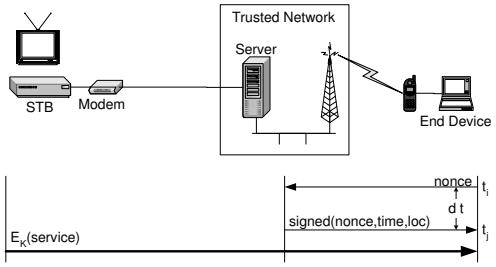


Figure 2: A time/location protocol (I)

This protocol requires access to a trustworthy interval timer to compute $\mathrm{dt}_{i,j}$. This requirement, however, can be removed if the content is provided in real-time with a logical interval timer embedded in the data stream.

This scheme depends on the allowed time delay, $\mathrm{dt}_{max}$. Choosing a value for $\mathrm{dt}_{max}$ could be difficult. If the threshold is too large then data will be able to travel out of the acceptable zone but if the threshold is too small then network jitter might cause blackouts for legitimate users. The choice of threshold becomes even harder if the network delay in the mobile network is variable. One possible solution to this problem is to allow the set-top box or the intermediary network to determine the threshold based on information received from the content provider and statistics from the trusted part of the intermediary network.

The disadvantage of this protocol is that it may be computationally expensive. The end device needs to generate a suitably random nonce, and the network needs to generate a signature which the end device needs to verify. This puts a strain on the end device, which may not have the computational power to verify signatures quickly; and the network, which may have to sign lots of messages quickly.

## 5 Reducing the Computational Load

To reduce the load on the end device, the following protocol, illustrated in Fig 3, moves the bulk of the computation to the set-top box. The set-top box also has access to a better source of nonces which may be derived cryptographically from the random keys that are used to scramble content. Typically these keys are changed several times a minute.

We assume that, in connecting to the network, the end device has been authenticated, both to the network as a device and to the set-top box as the intended recipient of the data stream. The protocol then proceeds as follows:

1. $ED \rightarrow STB$ :
   Request for usage criteria $\|$ $ID_{Service}$
2. $STB$ executes the DRM application and determines the usage criteria for the service.
3. $STB \rightarrow IN$ :
   Request $ID$ of location server nearest to $ED$.
4. $IN \rightarrow STB : ID_{LS}$
5. $STB \rightarrow CA$ :
   Request certificate for location server $\|$ $ID_{LS}$
6. $CA \rightarrow STB : \mathrm{Cert}_{LS}$
7. $STB$ executes $V_{CA}(\mathrm{Cert}_{LS})$ to verify $V_{LS}$.
   $V_{LS}$ is stored in the tamper-proof data storage area on $STB$.
8. $STB$ generates a random nonce $R_{STB}$.
9. $STB \rightarrow LS : R_{STB} \| ID_{ED}$
10. $LS$ generates $\mathrm{t}_i$ and stores $\mathrm{t}_i$ indexed by $ID_{ED}$.
11. $LS \rightarrow ED : R_{STB}$
12. $ED \rightarrow LS : MAC_{K_{S,E}}(R_{STB})$
13. $LS$ generates $\mathrm{t}_j$ and compares it with $\mathrm{t}_i$ indexed by $ID_{ED}$. If $\mathrm{dt}_{i,j} > \mathrm{dt}_{max}$ then $ED$ is geographically too far from $LS$ to provide reliable data.
14. $LS \rightarrow STB$ : time $\|$ loc $\|$ $MAC_{K_{S,E}}(R_{STB})$ $\|$ $S_{LS}($ time $\|$ loc $\|$ $MAC_{K_{S,E}}(R_{STB}))$
15. $STB$ then checks the validity of the signature provided by the location server using $V_{LS}$. This verifies the origin of the time and location data.
16. The $STB$ then verifies $MAC_{K_{S,E}}(R_{STB})$ to authenticate $ED$.
    If the MAC cannot be verified then $ED$ cannot be trusted. If the MAC is validated, then $STB$ can trust that the device communicating with $LS$ is $ED$, and that the data received has not been replayed.
17. The usage conditions (time and location) are then passed to the DRM application, which permits, or denies, delivery of the service ($ID_{Service}$) to $ED$ as appropriate.
18. $STB \rightarrow ED : E_{K_{S,E}}(Service)$
19. The protocol repeats from step 8 to step 17 at regular time intervals determined by the DRM application. This ensures that ED remains within the permitted location.

Again, this protocol ensures freshness by the use of nonces, nearness by the use of a time interval, entity authentication of the intermediary network by the use of a digital signature and entity authentication of the end device by the use of a shared key. The protocol requires the same measure of trust in the intermediary network as in the previous protocol but the trust in the user's end device is reduced, as is the computational load on this end device.

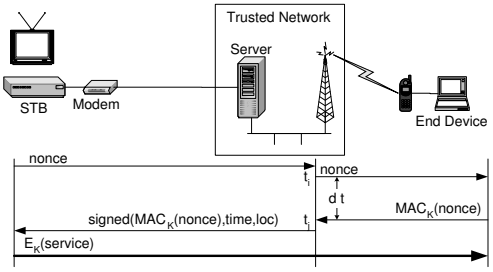


Figure 3: A time/location protocol (II)

## 6 Conclusion

Three methods for securely determining that time and the location of an end device have been presented: one that relies on trusted hardware and two software protocols collaborating with a trusted intermediary network.

An issue that has not been discussed is the effect on quality of service due to the increased computational loads. The protocols may also be vulnerable to denial of service attacks where an attacker may block or delay any of the messages causing the DRM software to believe that a legitimate user is in a blackout region.

The protocols could also place a large computational load on the intermediary network that signs the messages. A time-memory trade-off can be performed here, instead of signing a message each time, the intermediary network could negotiate a shared symmetric key with the message recipient and use a message authentication code rather than a digital signature. However in this case the intermediary network would have to store a symmetric key for each open session, and this could be equally undesirable. The development of protocols that place less demanding computational loads on the intermediary network remains an open problem.

## Acknowledgments

The work reported in this paper has formed part of the Core 3 Research programme of the Virtual Centre of Excellence in Mobile and Personal Communications, Mobile VCE, www.mobilevce.com, whose funding support, including that of EPSRC, is gratefully acknowledged. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE.

## References

[1] ISO, "Information technology – Multimedia framework (MPEG-21) – part 1: Vision, technologies and strategy", Technical Report ISO/IEC TR 21000-1:2001, International Organization for Standardization (ISO), Geneva, Switzerland, (2001).

[2] P. Bahl and V. N. Padmanabhan. "RADAR: An in-building RF-based user location and tracking system", In *Proc. IEEE INFOCOM*, **volume 2**, pp. 775–784, Tel-Aviv, Israel, (2000), IEEE.

[3] J. S. Erickson. "Fair use, DRM, and trusted computing", *Communications of the ACM*, **volume 46**, No. 4, pp. 34–39, (2003).

[4] European Commission. "Setting up the Galileo joint undertaking, council regulation (EC) no 876/2002", *Official Journal of the EC L 138*, page 1, (2002).

[5] United States' FAA satellite navigation website, http://gps.faa.gov/

[6] E. Gabber and A. Wool. "How to prove where you are: Tracking the location of customer equipment", In *Proc. Computer and Communications Security*, pp. 142–149, San Francisco, CA, USA, (1998). Assoc. Computing Machinery, ACM Press.

[7] S. Giordano, I. Stojmenovic, and L. Blazevic. "Position based routing algorithms for ad hoc networks: A taxonomy", In X. Huang X. Cheng and D.Z. Du, editors, *Ad Hoc Wireless Networking*. Kluwer Academic Publishers, Dordrecht, The Netherlands, (2003), to appear.

[8] Renato Iannella. "Open digital rights language (ODRL)", Technical Report ODRL-11, The Open Digital Rights Language Initiative, (2002).

[9] D. K. Mulligan. "Digital rights management and fair use by design", *Communications of the ACM*, **volume 46**, No.4, pp. 30–33, (2003).

[10] T. Roos, P. Myllymaki, and M. Tirri. "A statistical modeling approach to location estimation", *IEEE Transactions on Mobile Computing*, **volume 1**, No. 1, pp. 59–69, (2002).

[11] N. Sastry, U. Shankar, and D. Wagner. "Secure verification of location claims", In *Proc. ACM workshop on wireless security (WiSE '03)*, pp. 1–10, San Diego, CA, USA, (2003). Assoc. Computing Machinery, ACM Press.

[12] M. Valimaki and O. Pitkanen. "Digital rights management on open and semi-open networks", In *Proc. Second IEEE Workshop on Internet Applications (WIAPP '01)*, pp. 154–155, IEEE Computer Society, IEEE, (2001).

[13] X. Wang. "Digital rights management for broadband content distribution", In *Proc. 2003 Symposium on Applications and the Internet (SAINT 2003)*, page 5, IEEE Computer Society, IEEE, (2003).