Insider Attacks Enabling Data Broadcasting on Crypto-Enforced Unicast Links

André Adelsbach[1]* and Ulrich Greveler[2]

[1] Telindus S.A.
Security, Audit and Governance Services (SAGS)
2, rue des Mines, L-4244 Esch/Alzette, Luxembourg
andre.adelsbach@telindus.lu
[2] Fachhochschule Münster
Fachbereich Elektrotechnik und Informatik
Stegerwaldstr. 39, 48565 Steinfurt, Germany
greveler@fh-muenster.de

Abstract. Most wireless communication techniques rely on broadcast media on the physical layer, i.e., the actual signal can be received by any party in a certain coverage area. Furthermore, there are cable-based networks, such as HFC (hybrid fiber/coaxial) networks that use a shared transmission medium (coaxial cable) to bridge the last mile.
A common means to perform *secure unicast (point-to-point)* communication over such wireless or shared transmission media is by applying cryptographic protocols on higher layers of the protocol stack. As of today, a common assumption in the design and analysis of such communication protocols is that both end-points (user and carrier) behave correctly according to the cryptographic protocol, because they want to preserve security against outsiders who might be sniffing private communication of legitimate users. However, under certain conditions users may not be interested in protecting their unicast communication against outsiders. Instead, users may try to extend their communication power/resources by means of *insider attacks* against the communication protocol.
Such insider attacks pose new threats to providers of communication services and have, to the best of our knowledge, been neglected so far. In this paper we will discuss insider attacks against several communication systems that can break the unicast communication enforced by cryptographic means by the carrier of the communication infrastructure.

1 Introduction

Most wireless communication techniques rely on broadcast media on the physical layer, i.e., the actual signal can be received by any party in a certain coverage area. Furthermore, there are cable-based networks, such as HFC (hybrid

* The information in this paper is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The views expressed herein are solely the author's and should not be attributed to his employer or their clients.

fiber/coaxial) networks that use a shared transmission medium (coaxial cable) to bridge the last mile.

A common means to perform secure unicast (point-to-point) communication over such wireless or shared transmission media is by applying cryptographic protocols on higher layers: both communication end-points (user and carrier) set up a session key, which is then used to build secure (private and authentic) unicast communication by means of encryption and message authentication. As of today, a common assumption in the design and analysis of such communication protocols is that both end-points (user and carrier) behave correctly according to the cryptographic protocol, because they want to preserve security against outsiders.

However, if carriers have more power/resources in terms of bandwidth or coverage, users may not be interested in protecting their unicast communication against outsiders. Instead, users may try to extend their communication power/resources by means of *insider attacks* against the communication protocol. Such insider attacks pose new threats to these protocols and have, to the best of our knowledge, been neglected so far.

In this paper we present insider attacks, which can break the unicast communication enforced by the carrier of the communication infrastructure. We define a corresponding abstract communication model and sketch concrete instantiations that are deployed in practice: satellite ISPs, WIMAX ISPs and cable (DOCSIS) ISPs. We illustrate the effects of insider attacks mainly in terms of satellite ISPs, because here the effects of insider attacks are most striking: the user normally has a terrestrial link to the carrier and no means to broadcast data at all, while the carrier can broadcast its signals over huge footprints, covering millions of square kilometers. Our strongest insider attack may allow any end-user to make clear-text satellite broadcasts via the ISP's satellite, even if the downlink (data sent from the satellite to earth) is properly encrypted by the satellite ISP, thereby breaking the unicast communication structure enforced by the satellite ISP.

1.1 Outline

In Section 2 we introduce the general communication and attacker model and discuss state-of-the-art instantiations of this abstract communication model. Special emphasis is on satellite ISPs, as we consider them the most interesting scenario for insider attacks. In Section 3 we discuss several possible insider attacks against communication protocols that allow an insider attacker to broadcast messages via the ISP, although the ISP applies encryption. Section 4 we consider state-of-the-art communication systems and show that they are susceptible to our attacks. We conclude in Section 5.

2 Communication Models

Before going into the details of our analysis, we introduce the abstract communication model assumed in the remainder of this paper.

2.1 Abstract Communication Model

The abstract communication model is as follows (cf. Figure 1): the *carrier* of the communication infrastructure (also referred to as *ISP*), has a direct connection to the Internet and offers Internet connectivity to its *users*. We consider one of these legitimate users to be the *insider attacker*. The carrier/ISP communicates to its users via a physical broadcast medium or shared medium. As such, the communication (signals) from the carrier to its users can be received by any *outsider* in range of the broadcasted signals or having access to the shared medium. However, outsiders do not have access to the broadcasted message as it is normally encrypted with a key shared between user and carrier to preserve confidentiality against outsiders. Communication from the users to the carrier may be either via broadcast/shared-medium (visible to outsiders) or via a private communication link (not visible to outsiders).

Goal of the insider attacker is to make the carrier broadcast any message he likes, such that it can be received by outsiders.

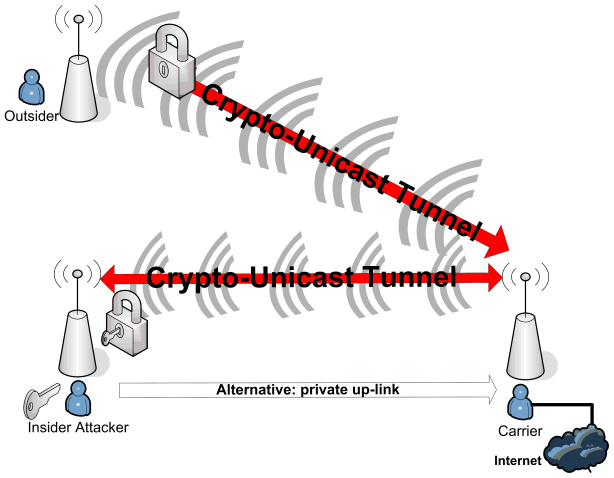

Fig. 1. Abstract Communication and Attacker Model.

Below we review wireless- and shared-medium-based ISPs and communication technologies that are state-of-the-art instantiations of this abstract communication model and, as such, susceptible to the attacks considered in our contribution. *Throughout the paper we assume that the insider attacker has read-access to its key-material, which in practice might require hardware tampering if the protocol is implemented completely in hardware.*

More specifically, we will review three types of ISPs: Satellite ISPs, WIMAX ISPs and Cable ISPs. As satellite-based ISPs have the biggest asymmetry in terms of coverage, we consider them the most attractive target for insider attacks

as discussed below. Therefore, we will mainly focus on satellite-based ISPs to illustrate these attacks. However, it is important to note that our attacks also apply to other wireless/shared-medium communication protocols.

2.2 Satellite ISPs

A satellite is a specialised *wireless* transmitter placed in terrestrial orbit for television broadcast, radio communications and data services, such as Internet access. Especially in low-infrastructure areas with limited physical/terrestrial networks, satellites provide high-speed access to the Internet. Some of the existing digital communication satellites transmit TV signals together with data communication so a user can use her dish and digital receiver for TV reception as well as one-way Internet access. Satellite-based Internet access comes in two flavours:

- One-Way with PSTN/ISDN up-link (asymmetric): In this lower cost option the satellite handles only the data downstream with outbound data travelling via modem/ISDN offering high download bandwidth[3] and a rather small terrestrial up-link bandwidth (up to $2 \cdot 64$ KBit/s).
- Two-Way with DVB-RCS up-link (symmetric): The more expensive two-way access requires a satellite terminal, e.g., a VSAT (Very Small Aperture Terminal), at the user's side and yields higher bandwidth for up-links (up to 2 MBit/s). Other types of return channels are under consideration, but are not broadly available yet (examples are *Return Channel Cable* and *Return Channel Terrestrial*).

Independent of the flavour (one-way or two-way), users and provider of satellite-based data services have highly asymmetric capabilities, both in terms of bandwidth (45 MBit/s downlink vs. 2MBit/s up-link) and in terms of coverage: the user normally has a terrestrial link to the carrier and no means to broadcast data at all.[4] The carrier, on the other hand, can broadcast its signals from the *exposed orbital satellite position* all over a huge footprint, covering millions of square kilometers and hundreds of millions receiver.

Below we consider the operation of an (asymmetric) one-way satellite ISP, because it is still more relevant. To illustrate how one-way satellite-based ISPs operate, consider the setting where a user fetches a file from a web server. A user establishes a small bandwidth dial-up Internet connection, e.g., using an ISDN line to some ISP (which is not necessarily the same as the satellite ISP).

[3] The current bandwidth (applying the Digital Video Broadcasting (DVB) standards [5]) offers a maximum data transfer rate of 45 MBit/s on one transponder frequency being shared among several users for the downstream. However the provider often limits the downstream bandwidth per user to a certain value, e.g., 1024 KBit/s.

[4] Even if the user had a VSAT being able to send data, this would not allow broadcasts with a comparable coverage, both because the transmitting power is significantly lower and, even more importantly, the dish is located on earth, allowing only to broadcast to a limited area (line of sight).

In order to initiate a download a request is sent through the dial-up line to an ISP proxy server (Fig. 2, Step 1), which relays the request to the desired destination (Fig. 2, Step 2). The reply coming from the server (e.g., a HTML page) (Fig. 2, Step 3) is re-routed by the satellite ISP so that it will not come back to the user's PC through the dial-up line. Instead it will be encapsulated together with the user's specific IP address into a signal based on the DVB standard and the ISP ground station relays it to the satellite (Figure 2, Step 4). The satellite broadcasts it and the user, having a satellite dish, a usual DVB card and a dedicated proxy software, decodes the response on his PC. The proxy software completes the TCP communication transparently for the application or operating system (Fig. 2, Step 5).

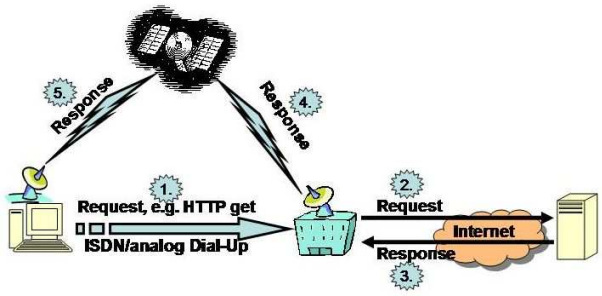

Fig. 2. How a satellite Internet connection works.

Due to the broadcast character of satellite signals, the signal dedicated for this user can be received by anyone in the footprint of the satellite.

2.2.1 Performance vs. Security Satellite communication links differ in several characteristics from terrestrial channels used for Internet communication [3]: Firstly, satellite communication links have a relative large latency, due to the large altitude of satellites.[5]

This large latency also results in a long feedback loop between sending a TCP packet and receiving the corresponding acknowledgement. Another important characteristic is that satellite channels have a higher bit-error rate than wired terrestrial networks.

As these characteristics would result in a significant performance degradation of the TCP protocol[6], performance enhancing techniques have to be applied

[5] According to [2] many satellites are located at the Geostationary Orbit with an altitude of approx. 36.000 km. This results in a propagation delay of at least 279 ms for one hop (station-to-satellite-to-ground).

[6] As delayed feedbacks and transmission errors are interpreted as signals of network congestion, TCP would automatically reduce the transmission rate. For more details see [3].

by the satellite ISP in order to achieve a satisfactory performance. Important examples for performance enhancing techniques are

- local TCP acknowledgements by the ISP's proxy (to speed up TCP slow start),
- local negative acknowledgements to improve error recovery,
- local TCP retransmissions from the satellite ISP to the client if a packet was lost on the satellite link (improve error recovery),

For details, we refer the interested reader to [2–4].

Satellite ISPs use a combination of performance enhancing techniques along this line to improve the overall performance of their Internet access. Moreover, application layer improvements, such as prefetching objects embedded in HTML documents that have been requested by a user, are in place. Unfortunately, these performance enhancements, implemented in *performance enhancing proxies (PEPs)* and operated by satellite ISPs, are not compatible with standard low-level security measures such as IPSEC because PEPs break the end-to-end semantics of a connection and the proxy interferes with the transport layer security measures, e.g., by sending pre-acknowledgements. Therefore, some PEPs also include proprietary security protocols to encrypt and authenticate communication.

2.3 WIMAX ISPs

The first WIMAX standard, known as 802.16-2001, incorporated a pre-existing standard, DOCSIS (Data Over Cable Service Inferface Specification). However, due to the different threat models of wired and wireless communication, security of the original 802.16-2001 standard failed to provide adequate protection. Johnston and Walker [7] discuss several security weaknesses of the original standard, some of which are:

- use of 56-Bit DES encryption in CBC-mode (cipher block chaining)
- lack of mutual authentication: base station does not have to authenticate itself
- no formally defined and weak authorization security association, paving the way for replay attacks

This security level had been considered sufficient, because 802.16 had originally been designed to be a *line-of-sight* point-to-multipoint communication system. As new features, such as mobility and non-line/near-line of sight (NOLS) were to be included in the standard, these security weaknesses became increasingly pressing and have been addressed by improving the 802.16 security mechanisms in the new version of the standard IEEE 802.16-2004.

The security sublayer consists of two main components: first, the *PKM (Privacy and Key Manangement) protocol*, providing network access control (authentication based on public key cryptography and X.509 certificates) and secure key distribution from the base station (BS) to the subscriber station (SS). Second,

the *encapsulation protocol*, defining a set of supported cryptographic suites and rules for applying these suites to encrypt (and authenticate) MAC PDU payload. Besides DES CBC packet data encryption, 802.16-2004 defines AES encryption in CCM (Counter Mode with Cipher Block Chaining MAC) mode as an additional packet data encryption algorithm.

3 Insider Attacks and Countermeasures

3.1 A General Definition

Classically, *insider attackers* are able to use a given computer system with a level of authority granted to them and violate their organization's security policy [11]. The insider attack is often considered as the primary human threat to computer systems since users that operate inside an organization have specific motives and legitimate access to systems that are detached from public networks. The commonly held views that most attacks come from the inside is a myth, though [10].

In the following we consider a different class of insider attacks, where the insider does not compromise a *system* from the inside of an organization but rather misuses a communication protocol for his own purposes, actively attacking security mechanisms established by the carrier.

3.2 What is the Motivation for Attackers?

There are several answers to this questions: first, attackers may want to break the unicast communication to get improved broadcast capabilities, which they do not have otherwise. This motive applies mainly to Satellite ISPs or Cable ISPs.

Second, even if the capabilities of users and carriers are more or less equal, an insider attacker may try to make the carrier broadcast illegal content, rather than broadcasting the content himself. This motive applies to all use-cases, including Satellite ISPs, WIMAX ISPs and Cable ISPs.

3.3 Why is this a problem for Carriers?

Again, there are several answers to this question. First, breaking the unicast structure may destroy the carrier's business model, because they may want to sell broadcasts at a significantly higher price than unicast communication. Loosing the ability to enforce unicast communication may destroy such business models.

The second reason is the carrier's potential liability for clear-text data broadcasts triggered by users. Even if the carrier disclaimed liability of such "unauthorized" broadcasts, these broadcasts may still significantly harm the carrier's reputation. Furthermore, doing broadcasts normally requires permissions by government, e.g. the FCC (Federal Communications Commission) in the U.S. By breaking the unicast communication insider attackers can misuse the carrier's

infrastructure to perform their own broadcasts without such permission. By operating such susceptible infrastructure, carriers may be held liable as well.

Moreover, according to "Data Retention" legislation carriers of telecommunication infrastructures are (or will be required)[7] to retain, amongst others, data necessary to trace and identify the *destination* of a communication. Given the perfect receiver anonymity provided by broadcasts, carriers can not even collect this data.

Finally, the ability to perform data broadcasts, having the same coverage as the carrier itself, may be used to attack other services, operated by the same carrier. Consider, as an example the satellite-based broadcast of Pay-TV. There are well-known attacks, referred to as *card-sharing attacks*, where a legitimate Pay-TV customer that paid for the TV broadcast, sends the required decryption keys to its peers. Today, the latter is mostly done via unicast Internet communication, which does not scale to large groups. However, as we will see below, insider attackers may misuse Satellite ISPs to broadcast Pay-TV keys using the same satellite infrastructure that distributes the encrypted Pay-TV content. This may significantly harm the business model of the Pay-TV provider and, indirectly, the business model of the carrier operating the satellite.

3.4 Insider Attacks on Crypto-enforced Unicast-Communication

Crypto-based communication protocols distinguish two phases in general:

1. first, the *key-agreement phase*, where a common session key is set up between user and carrier and,
2. second, the actual *encrypted transmission phase*, where the session key is used to encrypt messages before transmission over the broadcast/shared-medium.

Insider attacks may address both phases and we will consider them separately in the following sub-sections.

3.5 Leaking the key

Leaking the encryption key is always a straightforward option for an insider and there are several ways to leak the key material to outsiders. Besides sending the key material to the outsiders directly, it is also possible to distribute the session-key in a more advanced way.

[7] In Europe the so called "Data Retention Directive", issued on 15th of March 2006, will harmonise the data retention obligations of providers of publicly available electronic communications services or of public communications networks. This directive has to be adopted by national legislation in EU countries until September 15th 2007. In the U.S., according to the "Electronic Communication Transactional Records Act 18 USC s 2703(f)", ISPs have to retain records for 90 days upon request of a government entity.

- **Send the key to the outsiders via direct communication:** This is a straightforward way, but requires the insider attacker to communicate to each outsider individually. For large groups of outsiders, this seems to be infeasible. Furthermore, if the overall goal of the insider attacker is to preserve the anonymity of the outsiders (receivers of its communication), he must not communicate with the outsiders directly.
- **Publish the key in a newsgroup or electronic message board:** This is a straightforward way to publish the secret key, while preserving the anonymity of outsiders. However, it requires each outsider to access the published key.
- **Build a covert channel to broadcast the key via the broadcast/ shared-medium:** Here, the insider attacker may for example build a covert timing channel by sending/requesting (encrypted) data packets via the broadcast/shared-channel and encoding the key by following a certain timing pattern. Outsiders may observe the timing-pattern of these (encrypted) packets and reconstruct the key, allowing them to decrypt the packets afterwards. Although this approach seems viable, it certainly involves a certain overhead to communicate the key. On the other hand, it preserves the perfect receiver anonymity offered by broadcast channels and allows receivers (outsiders) to stay completeley passive and, thus, untraceable.

These methods are most efficient, if the communication protocol uses long-term secrets and there are no session-keys or session-keys are distributed via the broadcast/shared-medium. In this case it is sufficient to distribute the long-term secrets once. This applies to the following types of key-agreement:

- **point-to-point key update using symmetric encryption**: a long-term symmetric key is used to distribute all further short-lived (session) keys. Here, the insider attacker can leak the long-term key once, such that all further (session) key updates leak to outsiders if the key-update is done via the broadcast/shared-channel.
- **Key-predistribution schemes** are always vulnerable if the leakage of predistributed keys is sufficient for the outsiders to derive the short-lived keys. This is for instance the case for Blom's symmetric key pre-distribution system, the Otway-Rees protocol or the Needham-Schroeder public-key protocol.

Countermeasures. Since the intentional leakage of an agreed two-party session key to a third party by one of the parties cannot ultimately be prevented, countermeasures can only *complicate* the insider attacker's task, either by forcing the insider to communicate directly to the group of outsiders or increasing the data-rate necessary to distribute the key, or *deter* the insider from publishing his key. We consider the following countermeasures:

- **Use private channel for key agreement** The party intending to complicate the insider attack (the carrier in our setting) can use non-broadcast/non-shared channels (e.g., the dial-up connection in case of one-way satellite ISPs)

for key agreement purposes. Regarding satellite ISPs the dial-up connection should be favored for key-exchange. Therefore, outsiders can not benefit from knowing long-term keys if the session key is updated via a non-broadcast channel. Instead, the insider attacker has to distribute each key-update separately, as described above, or attack the actual key-agreement protocol (see below).
- **Increasing the rate of key-updates** also increases the rate in which the insider attacker has to send the updated keys to the outsiders. As the rate of key-updates increases, insider attacks become less attractive.
- **Include sensitive data** that the insider attacker probably does not want to disclose to the group of outsiders. If, as an example, the insider attacker is forced to include information, such as credit card numbers, account balances, date-of-birth or authentication credentials into his keys, he needs to disclose all this information to other parties that he might not trust in extensively. The need to disclose confidential data does then discourage insider attackers.

3.6 Insider Attacks on Key-Agreement

Besides leaking keys to outsiders, the insider attacker may try to attack the key-exchange protocol itself in order to always yield a *fixed key* that is known a-priori to the outsiders - so there is no need to communicate it to the outsiders. However, as two-party key establishment is inevitably insecure when the secret key is revealed by one of the parties after running the key-agreement protocol, two-party key-agreement protocols often do not address insider attacks.

Nevertheless, the setting has been considered before and Menezes et al. ([9], Chap. 12) refer to it as *key control*: "*In some protocols (key transport), one party chooses a key value. In others (key agreement), the key is derived from joint information, and it may be desirable that neither party be able to control or predict the value of the key*". Obviously, the inside attacker favors the setting where he (or the eavedroppers) can predict the value of a key.

Certain widely-used key agreement protocols have not been designed to prevent insider attackers and, therefore, should not be used in the communication models considered in this paper.

- **Shamir's no-key protocol**: Shamir's no-key protocol is a key transport protocol that does not require any shared or public keys and provides protection from passive attackers.[8]
  The protocol works as follows: the first party A selects the secret key K, chooses a random value a (co-prime to $p-1$) and sends $K^a \mod p$ to the second party B. When B receives $K^a \mod p$ he chooses a random value b and sends $(K^a)^b \mod p$ to A. Finally, A sends $K^b \mod p = (K^{ab})^{a^{-1}} \mod p$ to B, who can decrypt $K = (K^b)^{b^{-1}} \mod p$.

[8] As the parties do not share any shared or public keys, the protocol is not secure against active attackers as messages cannot be authenticated, which paves the way for man-in-the-middle attackers.

If the insider attacker starts the protocol he can always select a fixed key K, e.g., $x12345678$, known a-priori to the outsiders. Otherwise, if the carrier acts as A, initiates the protocol and selects a good key K, the insider attacker B can choose a fixed value $b = 1$, such that outsiders can intercept $K^b = K^1$ in the last step of the protocol. In the latter case the insider attack works only if the key-exchange is done via the broadcast or shared medium (e.g., WIMAX-, Cable- or symmetric Satellite ISPs).

– **Diffie-Hellman key agreement**: Let p be an appropriate prime and let g be a generator of $\mathbb{Z}_p^*$. The basic Diffie-Hellman key agreement works as follows: first, A chooses a random secret a and sends $g^a \mod p$ to B. Then B chooses a random secret b and sends $g^b \mod p$ to A. Now A computes $K = (g^b)^a = g^{ab}$ and B computes $K = (g^a)^b = g^{ab}$.
Here, the insider attacker A can always select a fixed "random" secret a, known a-priori to outsider outsiders. This allows outsiders to compute the agreed fresh key $K = (g^b)^a \mod p$ for every random secret b chosen by party B. Furthermore, the insider attacker may choose secrets a of small order or even $a = 0$. The former restricts the order of the overall key K (cf. Menezes et al [9]) and the latter always yields the degenerate key $K = g^0 = 1$.
The latter attack even works if the key-agreement is not performed via the broadcast/shared-medium, but e.g., via the dial-up connection in the one-way Satellite-ISP setting.
Note, that there exist checks to detect this kind of attack. The point is that these checks are not necessary to protect communication against outsiders, but they are mandatory to prevent insider attacks.

3.7 Insider Attacks on Encrypted Transmission

As discussed above, one possibility to break the crypto-enforced unicast communication is to *give the outsiders access to the encryption key*, either by distributing the encryption key or by attacking the key-agreement protocol in order to yield keys that are a-priori known to outsiders.

In addition to these attacks it is also possible to attack the actual *encrypted transmission phase*. The basic idea is *to make the carrier broadcast the desired messages as its* ***ciphertexts***. To achieve this the insider attacker may request specially crafted messages from the carrier, which, upon *encryption* by the carrier, result in the attacker's desired message. The carrier will then broadcast the attacker's messages as part of the encrypted payload. In this way, the insider attacker can make the carrier broadcast any message he likes. In the following we will consider this attack in more details.

Let m be a message, let E be the encryption function used by the carrier and let k be the (symmetric) encryption key. The "attack" depends on the following observation: for many encryption algorithms E it is straightforward to compute a function E^{-1}, such that

$$E(k, E^{-1}(k, m)) = m$$

holds for any key k and any message m.

Assuming that the carrier uses an encryption algorithm E having this property, our idea is as follows: the insider attacker, wanting the carrier to broadcast m, computes $d = E^{-1}(k, m)$ and requests a download of d (e.g., from its own server) via the carrier. The carrier fetches d from the insider attacker's server. Before sending it back to the attacker via the broadcast/shared-medium, the carrier encrypts d, which results in the desired *cipher-text* m, which the carrier finally sends over the broadcast/shared-medium. The overall attack is illustrated in Figure 3 in terms of asymmetric Satellite ISPs.

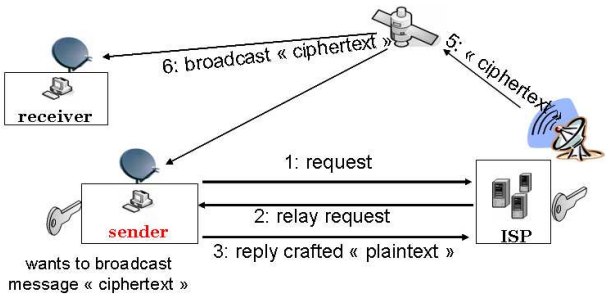

Fig. 3. Attack on encrypted transmission.

Below we consider some well-known, commonly used encryption algorithms and show that they are susceptible to this insider attack.

1. **Block Ciphers in Electronic Codebook mode (ECB):** In ECB mode of operation the message is divided into blocks that are encrypted individually by applying the block cipher to each block. For block ciphers there is an encryption function E and a decryption function D, such that the following holds: $m = E(k, d)$ and $d = D(k, m)$. Therefore, in this case, $E^{-1}()$ is identical to the decryption algorithm D, i.e., if an insider attacker wants the carrier to broadcast a certain "ciphertext" m, he simply has to request a crafted message $d = D(k, m)$ via the carrier.
2. **Stream Ciphers:** If the carrier applies a stream cipher, XORing the message with a pseudo-random key stream $PRNG(k)$ generated from k, the user may simply XOR the message m with the same pseudo-random key stream to get the desired crafted message $d = m \oplus PRNG(k)$.
   Practical examples of such stream ciphers are RC4 and block ciphers operated in Output Feedback (OFB) or Counter Mode (CTR).[9]
3. **Further Block Cipher Modes of Operation:**

[9] Basically, CTR mode computes the key stream by encrypting a nonce and a counter using the encryption key. As the nonce in CTR stays fixed throughout a connection and the counter value is increased deterministically, the attacker can predict the whole key stream once he knows the nonce.

Cipher-block chaining (CBC) mode of operation is illustrated in Figure 4 and the computation of E^{-1} is straightforward, as illustrated in Figure 5.

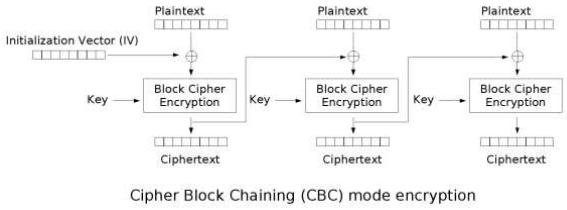

Fig. 4. CBC Mode of Operation (Source [8])

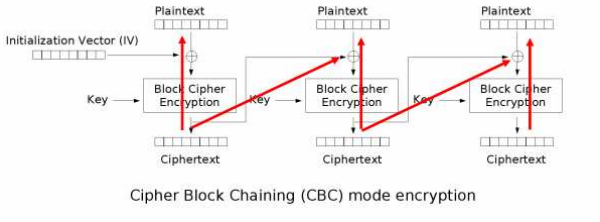

Fig. 5. E^{-1} for CBC Mode of Operation.

Remark: Note, that the IV required in block cipher modes of operation is also known to the insider attacker. It is either a part of the key, i.e., known before applying the attack, or sent to the insider attacker when starting the transmission. In the latter case, the insider can start its attack (crafting suitable packets) after he received the IV at the beginning of the transmission.

Remark: Note, that the insider attacker can even make the carrier broadcast messages that are specifically encrypted, such that only certain outsiders can read the message. For others (carrier or police), such messages look completely unsuspicious.

3.7.1 Countermeasures

Hardware implementations. A straightforward countermeasure is to fully implement the communication protocol, including both key-agreement and transmission phase, in hardware. A pure hardware implementation makes it harder for

the *average attacker* to manipulate the key-agreement protocol to enforce weak keys or to get direct access to the session key and mount the corresponding attacks on the encrypted transmission. Even if the hardware implementation could be attacked by patching the hardware, this attack would not scale as good as purely software-based attacks and, therefore, one may consider it kind of a countermeasure for low profile attackers.

Randomising Encryption. Let m be the message stream that has been requested by the user and that has to be forwarded to the user via the broadcast/shared-medium.

One possible countermeasure is to let the carrier add a random prefix x of fixed size to each message m before encrypting and forwarding it. By randomizing the encryption the insider attacker cannot craft messages that would lead to the desired ciphertext. On the other hand, this countermeasure is only effective for encryption schemes where a random prefix to the plaintext affects the whole ciphertext and it reduces the throughput of the down-link. Furthermore, it seems to be necessary that the carrier starts the encrypted transmission of $x||m$ only after m has been completeley received from the Internet, because otherwise the insider attacker could still adapt the tail of message m accordingly after he received the prefix x.

Fresh random message keys. Another possible solution is to let the carrier choose a fresh random message-key r for each message m and let the carrier send $E_r(m)||E_k(r)$. Here, again it is important that the insider attacker does not get to know r before message m has been completely received by the carrier. Otherwise, the insider attacker may again craft m, such that encryption with r yields at least partly to the desired ciphertext. This countermeasure seems to be effective, but again reduces the throughput and introduces computational overhead for the ISP due to extensive generation of message-keys. Furthermore, it may increase the latency, because the user cannot access a prefix of m before $E_k(r)$ has been received. We consider further analysis of these countermeasures as interesting future work.

4 Case Studies

4.1 DOCSIS and WIMAX

The WIMAX key management protocol (PKM) allows key transport from carrier to user. Similarly, the BPI+ (Baseline Privacy Interface (Plus)) of the DOCSIS standard involves a key transport protocol (BPKM - Baseline Privacy Key Management). As such, these protocols are not susceptible to insider attacks against the key-agreement phase, where the insider attacker enforces weak, a-priori known keys to be used. However, an insider attacker may still extract its key and distribute it to the outsiders.

DOCSIS 1.1 packet data encryption specifies 56-Bit DES encryption in CBC mode and according to DOCSIS 3.0 the CMTS (carrier) has to support 56-Bit

DES and 128-Bit AES, both in CBC mode. As such, Cable ISPs are susceptible to the insider attack against encryption as illustrated above.

WIMAX (802.16-2004) specifies 56-Bit DES in CBC mode as being mandatory to implement, whereas AES in CCM mode is not mandatory. Both CBC and CCM, which uses Counter Mode (CTR) for encryption, are susceptible to the insider attacks discussed above.

4.2 Satellite ISPs

Satellite ISPs offer the best gain for insider attackers, because users and satellite ISPs have completely different capabilities: the satellite ISPs sends its data through a satellite, being in an exposed orbit position, thereby being able to broadcast its signal over a huge area. Contrary, the user can only send its data through a terestrial uplink (point-to-point) or satellite up-link.

Although satellite signals offer no confidentiality due to their broadcast character, several satellite ISPs provide optional or no encryption at all [1], but solely rely on MAC-filters in DVB-cards or software drivers, which blind out transmissions of other users. However, similar to standard network adapters, DVB-cards can be put in a promiscuous mode, allowing anyone to receive the complete data downstream of a satellite. Given such weak non-cryptographic security measures, it is very easy to break the "enforced" unicast communication to achieve satellite broadcasts in practice.

There are PEPs that offer encryption, but their internals are not publicly specified. This makes it harder to analyse, which security mechanisms are actually implemented and whether these are susceptible to any of the attacks discussed above.

An analysis of software PEPs performed at Ruhr-University of Bochum [6] showed that it is quite straightforward to locate and extract the session key in the PEP software during runtime. Furthermore, the analysis (and searching the web) revealed that these PEPs use Diffie-Hellman key exchange (which might be susceptible as discussed above) and Blowfish encryption, which might be susceptible, if used in a susceptible mode of operation (see above).

5 Conclusion

We considered insider attacks against crypto-enforced unicast communication via wireless or shared communication media. Here the goal of an insider attacker is to break the unicast-communication that is enforced by cryptographic means.

We argued that it is important for carriers of such communication infrastructures, e.g., satellite ISPs or WIMAX ISPs to enforce unicast communication by means of strong cryptography. Current practice of some satellite ISPs that do leave users the choice to deactivate encryption or do not offer encryption at all, paves the way to misuse the carrier's broadcast capabilities, e.g., to broadcast copyrighted or illegal content over whole continents.

Even if the carrier encrypts its communication via the broadcast-/shared-medium we showed that it is not sufficient to focus on the classical outsider attacks when choosing key-agreement and encryption protocols. In addition to these classical attacks, it is crucial to consider insider attacks, as discussed in this paper. We showed that state-of-the-art communication systems such as WIMAX, DOCSIS or satellite ISPs are susceptible to these attacks, allowing insiders to break the crypto-enforced unicast communication and make the carrier broadcast arbitrary data. The effect is most striking in the case of satellite ISPs, where an insider attacker can make the carrier broadcast any message to the whole footprint of the satellite.

Acknowledgements

We like to thank Stephan Groß and Sandra Steinbrecher for interesting discussions regarding wireless communication protocols. Moreover, we thank Michael Steiner and Ahmad-Reza Sadeghi for providing their comments on insider attacks. Finally, many thanks go to the anonymous reviewers who provided detailed, insightful and critical commentary on this paper.

References

1. Andre Adelsbach and Ulrich Greveler. Satellite Communication without Privacy – Attacker's Paradise. In Hannes Federrath, editor, *Sicherheit*, volume 62 of *LNI*. GI, 2005.
2. M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, and J. Semke. Ongoing TCP Research Related to Satellites. Internet Request for Comment RFC 2760, Internet Engineering Task Force, February 2000.
3. M. Allman, D. Glover, and L. Sanchez. Enhancing TCP Over Satellite Channels using Standard Mechanisms. Internet Request for Comment RFC 2488, Internet Engineering Task Force, January 1999.
4. J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. Internet Request for Comment RFC 3135, Internet Engineering Task Force, June 2001.
5. DVB Project. Dvb project homepage. `http://www.dvb.org`.
6. T. Haskes and J. Kopperschläger. Analysis of a satellite isp client. unpublished bachelor theses Nov 2005.
7. David Johnston and Jesse Walker. Overview of ieee 802.16 security. *IEEE Security & Privacy*, 2(3):40–48, 2004.
8. Lunkwill. CBC illustration. Wikipedia, `http://en.wikipedia.org/wiki/Image:Cbc_encryption.png`, 2004.
9. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7.
10. E. Eugene Schultz. A framework for understanding and predicting insider attacks. *Computers and Security*, 21(6):526–531, October 2002.
11. T. Tuglular and E.H Spafford. A framework for characterization of insider computer misuse. Unpublished paper, Purdue University, 1997.