Escrow Encryption Systems Visited: Attacks, Analysis and Designs

Yair Frankel[1] * and Moti Yung[2]

[1] Sandia National Laboratories, Albuquerque, NM
[2] IBM T. J. Watson Research Center, Yorktown Heights, NY

Abstract. The Escrow Encryption Standard and its realization – the Clipper chips – suggest a new type of encryption scheme. We present a few basic and somewhat subtle issues concerning escrow encryption systems. We identify and perform attacks on the actual Clipper and other recent designs (fair cryptosystems, TIS software escrow, etc.). We review requirements and concerns and suggest design approaches to systems with desired properties of key escrow.

1 Introduction

The Escrowed Encryption Standard (EES) and more specifically the Clipper chip family [18, 8] is the U.S. government's approach to technically bridging the gap between a users' need for cryptographically strong privacy and society's need for protection against criminal behavior. When key escrow was introduced, it resulted in heated discussions about the Clipper program (which includes recoverable keys from two trusted escrow agents when a court order is issued). The perspective we take here (from a technical stand point — leaving our personal views aside) is that there is a need for both criminal investigation mechanisms (fully handled by the judicial system) and full individual privacy (as protected by the Constitution). Further based on the above starting point, we advocate a sound approach and attempt a prudent analysis that covers both aspects.

The obvious threats to escrow systems considered already are those of dishonest escrow agents (which is why Clipper uses two agents) and a dishonest user (which is the purpose of the law enforcement access field (LEAF) in Clipper). We will assume here that Clipper's classified encryption algorithm, SKIPJACK, is strong enough to protect the user's privacy due to [6]. However, this may not be enough. It appears to us, in light of the findings of this paper, that the concept of escrow encryption for law enforcement purposes is not a well understood one. We analyze requirements and ask: are there any inherent subtle problems with existing systems? and, are there necessary security mechanisms lacking in existing designs? At the same time we attack existing designs (like the Clipper and fair cryptosystems). We hope that our findings and analysis will allow the escrow encryption systems designers and users to build better systems which satisfy both user's and law enforcement's needs.

* Research was performed while the author was at GTE Laboratories Incorporated.

D. Coppersmith (Ed.): Advances in Cryptology - CRYPTO '95, LNCS 963, pp. 222-235, 1995.

Attacks: As part of our analysis we raise several concerns and attacks. We show that the actual Clipper chip based system allows a scenario where the potential criminal can attack and somewhat choose whose device key must be opened to read the criminal's own messages by *squeezing* LEAFs. Squeezing also enables an attack by two interoperable rogue entities within two way communication systems; its strength is that law enforcement cannot catch the fact that the entities do not follow the Clipper specification. We also demonstrate that a public key system on its own (e.g. as in Crypto'92 fair cryptosystems) is not good enough for key escrow. Moreover, in environments like the Internet with both fair cryptosystems and Clipper, two collaborating parties can spoof two legitimate users and can "frame" them, by faking their own session to "look like" it is between the legitimate parties.

Methodology: We take a three step approach in our investigation. 1) We identify basic design issues in escrow encryption cryptosystems and provide questions we believe designers should ask in determining requirements. Rather than speculating on what is needed, we use official documents as systems specifications. We are guided by the FBI requirements [17] (derived from the telephony law) and by the administration goals as stated in press releases [32]. We also follow the tradition of cryptographic design. 2) We then relate our questions and concerns to proposed technologies (e.g., Clipper [8], fair cryptosystems [25], TIS software key escrow [2], etc.), and 3) we discuss threats and demonstrate the attacks mentioned above on actual systems and designs when they fail to solve the basic issues. We also extract basic design issues from the analysis; further designs we deduced will be discussed in [19].

Overview– escrow encryption systems: Let us first review the basic key escrow systems which provide a third party – the escrow agency – with the capability to decrypt sessions of a registered user (device). To provide for added trust and security, the escrow agency is often treated as a distributed function (multiple entity collaboration). Most of the proposed schemes are based on one or combination of the following two approaches.

- The LEAF approach (Clipper is the prototypical system for this approach) transmits an encrypted session and key identifier with the ciphertext. Law enforcement, given the key identifier, obtains from escrow agents the key associated with the identifier and decrypts the encrypted session key [8].
- A threshold decryption [13] approach provides agents with a share of a user's key. The user encrypts a message and the agents are able to decrypt the message using their share of the user's key. Micali's fair cryptosystem [25] is the prototypical system for this approach.

Organization: Section 2 describes issues and attacks concerning identification of entities and proof of compliance that assure that messages can be retrieved by law enforcement. Section 3 describes specifically the squeezing attacks realization. Section 4 analyzes key management issues while Section 5 presents attacks on public-key based systems. Section 6 analyzes the system operation stage and general issues.

2 Entity Identification and Compliance Certification

Wiretapping over phone lines does not provide for strong authentication of the user and messages, so it is often used only to aid in collection of evidence rather as an evidence itself. Thus, strong authentication is not a wiretapping requirement. Yet, identification in escrow encryption systems is a more subtle issue than in a typical cryptographic protocol. Identification determines a binding between the device/ user and a key to be taken off escrow. The need for identification, therefore, surfaces naturally. Regarding this function we ask:

Issue A: Entity identification/ message authentication issues

1. Must there exist a secure binding between a sender (and/or receiver) identity with the ciphertext (in a general network environment)?
2. Should users insist that their key gets opened only when they are part of the action (session)?
3. Is it possible and practical to provide message authentication?

The Clipper chip transmits a chip ID field (as part of the LEAF) in order to provide law enforcement with the information to determine which key needs to be removed from escrow. Also fair cryptosystems and other suggested escrow encryption systems must provide a key identifier for a similar reason. Such a field has a potential of establishing the source or destination of the message. Is this required or does this provide law enforcement with a stronger tool than current wiretapping provides? In fact, this is required and assuring that wiretapping is performed w.r.t. the specified subject is a must (by the FBI's own specification)! We observe that the FBI specifies in Requirement 5 (Verification Information) [17]: *"Law enforcement agencies require* (1) *information from the service provider to verify the association of the intercepted communications with the intercept subject, and* (2) *information on the services features subscribed to by the intercept prior to intercept implementation and during the intercept"*

The FBI requirements further state: *"Specifically, court authorities require law enforcement agencies to verify that the communications facilities or service being intercepted correspond to the* **subject or subjects** *identified in the lawful authorization."* In fact currently for voice wiretapping the notion of "minimization" rules is used, namely the listening has to concentrate on the subject only (when identified by its voice characteristics) and not on its line that may be used by others. Moreover the U.S. Department of Justice (DOJ) in its letter discussing authorization procedures for release of encryption keys states that federal agents may obtain a certificate which shall [9] (requirement 3f): *"specify the identifier (ID) number of the key-escrow encryption chip providing such encryption ..."*. This, however, does not seem to be an absolute requirement[3].

[3] It states "...non-compliance with these procedures shall not provide the basis or motion to suppress or other objections to the introduction of electronic surveillance evidence lawfully acquired.

For wiretapping of data we need other means for identifying a subject. Further, for general network environment we cannot always rely on source identification (sources may be multiplexed). This could be one of the roles of the LEAF. However, it is crucial to point out that the law enforcement requirements [9, 17] are not in agreement with Clipper.

Claim 1 *There is an attack showing that the LEAF field in Clipper does not determine the identity of the chip which encrypted a message. Moreover, criminals can squeeze an honest person's LEAF in order to force escrow agents to open the honest person's escrow key.*

The above should be a concern to, both, users and law enforcement. If a sender is not identified in a secure fashion then he/she may be *framed* (e.g., suspected of an action that one did not do). Even if the government does not intend to make an implication about binding of user to device (which we expect is the government's current policy) this does not prevent the defendants from making such implications to generate an alibi by "proving" that they were in the locality of the supposed encrypting chip during the crime when they actually were not. Thus, the implication of identity and binding of message to chip is a dual-edge sword abusable by both law enforcement and criminals (this borrows a terminology from [32] where encryption is described as "dual-edge sword"). We believe that the fact that binding of chip identity and a ciphertext cannot serve as an alibi must get known to the courts!

Claim 1 also tells us that **users who have no part in some illegal action (whether knowing it or not) may have their keys opened and messages read.** This is specifically easy over packet switching networks (the Internet) and wireless networks. Such possibilities may defeat the purpose of the escrow agents as protectors of individual privacy. We believe that showing that in some cases wiretapping is possible only if a potential violation of privacy is allowed must be made clear to the courts!

One may claim that Clipper is intended only for voice (which provides for user authentication)– but we note that data seems to be an important goal of the key escrow program. In fact, a data processing device – the Tessera crypto card — is a PCMCIA version of Clipper [26]. Moreover, [9] specifically pertains to other electronic surveillance.

Message authentication (not included in Clipper) is more costly than identification. Such operation inside the tamper-proof device will enable user protection against law enforcement "planting" messages in wiretapping (and will thus make information gathered more admissible in courts, since in this case no one can tamper with the ciphertext). The tamperproof devices will insist on proper authentication prior to decryption. This, however, requires tremendous overhead of processing each message (in voice and real time applications we may be able to authenticate only parts of the message).

The issue of the tamper proof insisting on verification brings up the question of compliance certification: the fact that the system's operation is assured only when the user comply with its requirement (which we next discuss briefly):

Issue B: On compliance certification

1. What level of certification of the usage and compliance is necessary?
2. What level is technologically possible and practically acceptable?

It is always possible to bypass an escrow system's algorithm or use another encryption within it. There is nothing that can be done about that. Defining compliance (what is correct usage) and providing a certification mechanism (and by whom?) is a difficult problem; at least we can say that compliance should assure that: *a user employing solely the "encryption portion of the mechanism" for assuring secrecy, should also enable law enforcement readability of the message*. How hard is it to bypass compliance in Clipper? In fact, Blaze has shown that it is possible due to short integrity check (16 bits), and that collaborating rogue entities can avoid sending the LEAFs (where they can be caught by law enforcement). In fact, in Section 3 a self-squeezing attack is described. It can be viewed as a strengthening of Blaze's second attack [5]. The strengthening enables misbehaving entities to avoid being caught as such.

Claim 2 *There is an attack showing that compliance is not possible in Clipper given two rogue parties (even if the LEAF contains the identity of the parties' chips and proper integrity check).*

In theory, zero-knowledge techniques can certify compliance. In practice however it is hard; it is much easier (but costly) with tamper-proof devices checking for certification as part of their function. One suggested design in [25] assures that a key is properly distributed. This may not be enough for operation time. The solution in [25] for operation time is to make the use of other keys illegal (which is a non-technical issue), but a criminal may just as well risk violating the law about the key in order to hide the perhaps much bigger violation which is the actual crime discussed. Some solutions assure that some sampling of the use can be done by authorized tapping to verify that proper text is being sent [10].

3 Squeezing Attack: How to Use Other People's LEAFs

We now realize the attack of Claim 1. The compliance certification in Clipper attempts to assure Law Enforcement that a user receives the LEAF from the other user before the ciphertext can be opened as required by [18].

Our experiment: We performed the squeezing attack by experimenting with an actual Mykotronx MYK-78T which performs SKIPJACK in Output Feedback (OFB) mode [27], it was part of a board constructed in GTE labs. The chip works in the following manner. For encryption mode: an external session key is input and then the device outputs the IV and LEAF. Afterwards, encryption using SKIPJACK in OFB mode with the session key and IV can be performed. For decryption mode: the session key, IV and LEAF must be input and pass the identification test (to test for compliance) before decryption is allowed. (Observe the relationship between identification and compliance used here).

We had two MYK-78T chips which were given a session key KS and in the spirit of [18] the IV/LEAF tuple was transmitted with any ciphertext. Let the LEAF generating function be H. Chip A first generated the IV=IV_A and a $LEAF_A$=H(IV_A,KS,..) and then encrypted a message. The attacker's chip B having received the session key KS earlier, and the $IV_A/LEAF_A$ tuple and the ciphertext from A was then employed.

We attempted that B would encrypt its own message (ciphertext and $LEAF_B$) and then will decrypt the modified message (same ciphertext and $LEAF_A$). So B's output must be consistent with $IV_A/LEAF_A$ tuple to be concatenatable to B's ciphertext. However, B can not use encryption mode to accomplish this since in encryption mode the MYK-78T chip chooses its own IV=IV_B internally. This problem was, however, resolved by using chip B in decryption mode. So that we could enter in the IV_A manually as IV_B (assuming the IV_A is chosen by the supposed "other party" to the "legal session"). We were able to verify our test showing that B was able to conversed with itself, using A's IV_A, replacing $LEAF_B$ to $LEAF_A$ before transmission. B then decrypted this transmission! To encrypt in OFB mode, an input of all 0 bits is input to the decrypting chip to retrieve the encrypting string (pad). This string can now be exclusive-ored with the plaintext to generate the ciphertext. Note that in ECB mode (as is expected to be provided in the MYK-78E which is in development), the IV problem is even simpler (as IV is ignored).

Discussion: What environment will enable our attack? In a typical attack, the LEAF will be squeezed from one device and be served in a session between the attacking device and a third party; similarly two devices can be attacked by two coordinated attackers. We used manual key distribution, RSA public key [30] can be used just as well – the common session key is used by both sides (as EES requires). If Diffie-Hellman key exchange were the approach then the attack becomes more complex, unless the Diffie-Hellman key exchange is used as a master key to encrypt session keys which again simplifies the problem. The difficulty in the attack is that the session key can not be uniquely controlled by the bad guy(s) since the good guy has an effect on the choice of keys when they start their session. For an attack to work in this case, the third guy has to be an accomplice and then again, both of the bad guys can use the same LEAF (i.e., the good guy will appear as both the sender and receiver in this case!); in addition, the key and LEAF must be shared outside the box as in MYK-78T and not under the tamper resistant device. A squeezing example is of two rogue government officials that talk with the president and the vice president, and employ their partners' LEAFs in an illegal private session between the two of them. Another example is a criminal organization that opens a 900 phone service and gets sessions (and LEAFs) of numerous privacy-conscious citizens which can be squeezed and used for simultaneous crime-related sessions.

Self-Squeezing- strengthening Blaze's attack: In one of Blaze's attacks [5] two rogue entities manage to avoid sending LEAFs. It is important to note that in telephony (two way communication) both parties must be rogue since [18] requires that both parties send a LEAF and use the same session key. (We, thus,

feel that this attack may be more important than what is actually believed). In his attack, two entities can bypass the system in a real-time fashion and drop the LEAF from communication. When getting wire-tapped, the two entities are caught. Using squeezing they can do so and their communication will look like they follow the compliance certification. All law enforcement will notice is that the messages sent are randomly looking while they are not (the entities may claim they were merely testing their devices and they can reconstruct the randomly looking messages!). To do this, the rogue entities generate two keys in the key exchange phase (one is the key and the other is, say, a derived key which is a result of a one-way function application over the key). They each generate a LEAF using the derived key (call it a derived LEAF), then use the actual key to generate encryption and true LEAF, and before transmission just replace the generated true LEAF with the derived LEAF as in squeezing. Before decrypting, a user replaces the sent derived LEAF with its own (true) LEAF. This operation is done in real time. Note that this attack is what is relevant to two way communication, is independent of the size of the LEAF and its fields, and cannot be avoided by disabling devices that fail too many times (a measure suggested against other attacks of Blaze).

On prevention: The squeezing attack on the LEAF could have been prevented. One possible design would be to have the session key be a function (hashed using a globally chosen pseudorandom function known only to devices) of the sender and receiver id (ID-based key distribution). Compliance could be verified under the protection of the tamperproof device. This assures that the key is used between two specific users and not others. To protect against self-squeezing a stronger measure is needed (message and message originator and receiver authentication).

4 Analysis of Key Management Issues

Key management is a major issue in key escrow. We ask:

Issue C: General key management issues

1. Who generates the user's keys?
2. How active should the user, escrow agent and other government agents be in the key distribution/management process?
3. What is needed for international escrow systems? How to make an international escrow when countries may not trust each other?
4. How is the key disseminated to escrow agents and how do we prevent cloning of keys?
5. Is it necessary to share keys of users.

The purpose of an escrow system is to allow strong cryptography, therefore weak keys are unacceptable. Thus, the user may wish to play a strong role in the process of generating escrow and session keys (as in Clipper) whereas device generation of keys (as in Capstone) may assume reliance on the device designers (e.g. to use a proper pseudorandom generator).

Too much involvement of or shepherding by the escrow agents in various stages is highly undesirable (in particular when hardware is used). In [4] law enforcement gets involved in session key generation in order to provide a means to restrict law enforcement in time; such context limitation is needed. Also, international key escrow may require agents trusted by mutually distrusting countries and a key hierarchy for escrow.

We next note that **key cloning attack** may cause illegal wiretapping and may be possible by fooling the court. The scenario allows an entity who is a "an intentional future suspect" to copycat a key of another user (say in [25] system). Then when this entity misbehaves, a court may issue a wiretapping order, which will enable law enforcement to listen to the honest user as well (due to the cloning). Consider the espionage potential of such an attack in the context of an international key escrow system and how the international court can be fooled.

We ask: should user registration involve (the typically complicated) sharing of its key? In fact, by employing tamper-proof devices at the escrow agents we designed a technique where the user's key is not required to be shared to agents. The technique of *protected function sharing* has numerous other applications and is discussed in [19, 20] and briefly in Section 6.

4.1 Tamperproof devices verification

Clipper uses tamperproof devices created by the government to hide the secret algorithms encoded in the chips and to some degree to enforce compliance. The main user concerns with a system based on the use of tamperproof devices are:

Issue D: Users' concerns with tamperproof encryption device

1. How to verify that the "correct" algorithms are encoded? and how to minimize potential problems in a device?
2. What prevents cloned devices?

Correctness and uniqueness are valid concerns of users when they may not trust the builder of the equipment. This is especially true when some of the algorithms encoded are classified and correctness is therefore ill-defined; the problem of Capstone internal algorithms not being trusted is mentioned above (which justifies independent key distribution). Auditing by independent trusted entities may provide some assurances of the integrity of the devices and procedures.

5 Attacking Public-Key based Systems

5.1 Public-key issues

Public key reduces the number of user keys and appears to be the method of choice for software escrow encryption systems for this reason. We note that due to e.g. [24, 14], private key systems protected under a tamperproof devices exhibit properties of public key. We ask:

Issue E: Public keys

1. Which keys should be opened by escrow agents?
2. How can the public key be abused by the user?
3. Is ID-based public key useful?

It appears to us that the use of public keys for escrow encryption is not well understood. Indeed with any "pure" public key approach to provide for escrow encryption [25, 13, 10] implies that opening ciphertexts requires a situation that is as unacceptable as the U.S. postal service having the authority to open one's mail under the suspicion that criminals are sending, say, mail to people in the neighborhood. Thus, naturally, we have to augment public-key techniques when used in the context of key escrow systems. We specifically claim about public key escrow systems that:

Claim 3 *Fair cryptosystems are not fair.*

Discussion: Fair cryptosystem discusses the monitoring "of user's suspected of unlawful activities" yet it achieves only the monitoring of messages to and not **from** suspected criminals. In an El-Gamal based Fair cryptosystem, a user A has a public key Y_A where $Y_A = g^{X_A} \bmod p$, X_A is the shared private key. To send a message in a session the criminal B chooses a random element R and sends to A $PUB = \langle K = g^R, C = Y_A^R * M \bmod p \rangle$ and A can recover the message by computing $M = C * (K^{X_A})^{-1} \bmod p$. For the law enforcement to get the message from the public information available PUB, they have to remove A's key X_A from escrow. Thus, fair cryptosystems *require* the opening of the keys of individuals receiving messages from criminals rather than opening the keys of the suspected criminals sending the messages. A criminal may get in sessions with as many as possible users (preferably, high government officials and secret service officers) which will enforce opening of all their keys. ♣

Potential abuses in the stage of key generation and distribution are known in traditional systems and apply also to key-escrow systems. The potential use of "law enforcement trapdoor" in the key exchange phase is suggested in [4].

We note that Desmedt [11] presented a method to enforce compliance by the agents generating a public key based on the owner ID. (This is a novel variant on ID based cryptosystem, the ID association is visible to the escrow agents). With ID-based systems, the receiver is traceable by the public-key used, as he noticed. As mentioned above, receiver traceability is not enough to assure fairness in wiretapping in public-key systems.

5.2 Spoofing attacks

We now assume that a public-key system is not used "as is" in escrow systems since the key is asymmetric and the lack-of-fairness attack. Rather we assume the public-key is employed for session key exchange. As advocated in [1] a designer should check what may go wrong with its system in an operational environment and not merely treat it as a separate component. So, here we consider the Internet, as an example of a vulnerable setting representing potential problematic

operational environments. The attackers are a pair of accomplices $\tilde{S}$ and $\tilde{R}$ who will spoof two honest users $\bar{A}$ and $\bar{B}$. $\tilde{S}$ sends from $\bar{A}$'s output queue (spoofing operation) a message routed to $\bar{B}$ which can be routed from $\bar{A}$ to $\bar{B}$. $\tilde{R}$ can read and remove the message from $\bar{B}$'s input queue (spoofing operation, again).

Claim 4 *Spoofing is easy with public-key based key escrow and the Clipper.*

Discussion:

Let $\tilde{S}$ and $\tilde{R}$ spoof $\bar{A}$ and $\bar{B}$.Consider a key-exchange based escrow encryption system. To dictate a key K, $\tilde{S}$ puts in $\bar{A}$'s output queue $< K^{e_B} \bmod n_B >$ where $< n_B, e_B >$ is $\bar{B}$'s RSA-based public key. This enables $\bar{B}$ to get K (if it were the case that $\bar{B}$ gets this message) and this convinces wiretapping law enforcement that this is a key from $\bar{A}$ to $\bar{B}$. In the message header (IP header over the internet) $\tilde{S}$ puts the value K itself in some predetermined (unencrypted) positions inside the message ID. $\tilde{R}$ has access to $\bar{B}$'s input queue, he reads the message and given K checks for the fact that it is also transmitted via the key distribution packet (this checking is possible using a filtering software such as those used in firewalls). If this is true, then $\tilde{S}$ and $\tilde{R}$ use K for their private conversation, while observers on the line (law enforcement) assume this is $\bar{A}$ and $\bar{B}$'s session which they cannot open since as far as they know and check, two honest users are following the law by using the key exchanged using the escrowed key (by the "minimization rule" of wiretapping they cannot listen to the honest users).

For example, if Fair cryptosystems are used for this key distribution, then under spoofing, they do not, in fact, enable opening of messages sent for criminals as claimed. Criminals who are good hackers and are well coordinated may enjoy total freedom as the network seems to be used only by seemingly honest users. Since compliance is provided in hardware for Clipper, first a LEAF has to be squeezed and then the above spoofing is possible. ♣

This claim shows that authentication, e.g., a digital signature protection, may be needed with key escrow system (like Fair cryptosystems) to assure the source liability to its messages and for prevention of spoofing. This is needed in any cryptosystem in such an environment, but seems crucial in escrow encryption to assure law enforcement function. Thus, one should not view popular signature providing software (like the available PGP and Privacy-Enhanced-Mail) as competition to key escrow, but rather as a necessary complementary component in environments like the Internet. Independently, [23] studied software escrow encryption systems and made similar observations about their weaknesses. One must, however, be careful to separate encryption and the signature mechanisms (which is not done in the popular systems) as the signature key should not be subject to escrow [20, 19]. We also note that with software-only escrow encryption using a public key system which simply simulates the Clipper system as in the TIS [2] one inherit the problems of both systems (and more).

To design software-based approach which has a compliance certification component and spoofing-thwarting capabilities one may rely on on-line servers and on-line proofs of compliance, and trusted servers that are not spoofable and can perform key-translation functions [20, 19]. Dealing with such compliance and

possible spoofing issues, indeed, looks very much related to subliminal channel questions [31]. This is a hard channel to prevent and deal with. (It was noticed in the context of key distribution in key escrow systems first in [22]). Thus, generally dealing with the worst case of spoofing seems hard.

6 System in Operation and General Issues

The additional parties (escrow agents, the court and law enforcement officers) and their capabilities increase the complexity of operation, and present potentially new abuses, attacks and mishaps in escrow systems cryptosystems. This introduces new subtle issues, some of which we discuss.

Issue F: Threats to operation

1. How should wiretapping be made limited in time and in context?
2. How are the traditional attacks on systems: e.g., chosen messages and chosen ciphertext attacks (see [28]) apply to escrow encryption systems?

Time and context limitation can be assured using various cryptographic techniques and operational procedures depending on the technology used (e.g. limiting the life-time of keys discussed above). However, the question also applies to off-line tapping (i.e., of recorded information) where enforcement of restrictions may be harder. The Clipper program uses strict operational procedures as required by various U.S. and state laws and statutes. An interactive protocol in which the law enforcement agents are active in session key generation is used in [4]. This allows for the protection of past messages. The notion of *balanced cryptosystems* where agents open communications based on a per message granularity (rather than a per device/ session key one) is a method to limit context and time of wiretapping (consistent with the "minimization rule" policy [17]) that was introduced in [10, 12]. Using [19, 20] it is possible to achieve this with Clipper by observing that one can first decrypt inside an escrow agent's tamper-proof device without revealing the decryption key or cleartext, and then output a share of the decryption (cleartext) rather than the actual decryption. The device contains the user key and decrypts the ciphertext to obtain message m, it also has pseudorandom function shared among devices. Based on the device ID and the shared randomness rather than sending m, m's share is sent out. This technique can be further improved by externally storing users' keys and techniques from [10]. Finally note that when considering traditional cryptographic attacks, they may now be against additional parties such as the court system, escrow agents, law enforcement agents etc.

General system issues apply to key escrow systems: reliability, efficiency, computing resources and technologies, etc. We next ask about such issues:

Issue G: General System Issues:

1. Is there a possibility of a software-only solution? How is it different from the Clipper approach? What technologies are required?

2. How can we assure reliability and availability?
3. Does availability of low price key escrow devices imply overall ease of wiretapping?
4. How does the type of application affect the technology used for escrow encryption? For example, how we make key escrow for storage channels (rather than communication channels)?

Having any system with hardware devices or purely in software is a technological tradeoff. As the vice-president has told the press, future key escrow technologies will be oriented towards software and data processing. Hardware seems to ease the design, in particular easing compliance certification (as mentioned above).

The notion of reliability and availability is an important general systems issue especially in emergency applications such as for ambulance and police services (availability vs. security in banking systems is discussed in [1]). The system must be available as denial of service may limit the usage and open new liability issues.

Claim 5 *The Clipper program puts compliance over availability.*

Discussion: As mentioned in [5, 8], a response to Blaze's attack on the 16-bit LEAF is to internally count bad LEAFs up to an upper bound $U << 2^{16}$ and then to disable the device. Now, a bad guy can flood a device with bad messages and disable it. This implies that the suggested remedy to Blaze attack may make Clipper unsuitable for sensitive high availability applications. ♣

On the issue of real-time, we observe that the FBI specifies in Requirement 2 (Real-time, Full-time Monitoring) [17]: *"Law enforcement agencies require a real-time, full-time monitoring capability for intercepts."*

Claim 6 *Cheap Clipper chips may reduce the feasibility of the Clipper program for real-time Law Enforcement.*

Discussion: The government may try to sell key-escrow systems for a low price to assure their popularity and enable simple key escrow program. This in turn, may imply that bad guys can afford a multitude of devices for relatively low investment of money. They may use them in an interleaving fashion. This will require excessive coordination/ involvement of the escrow agents and the courts. This may make real-time wiretapping a very hard task. ♣

We also note that the type of applications has always influenced how we do encryption for it. There appears to be a difference between communication channels (which are somewhat real time applications like e-mail and telephony) and storage channels where information is kept long time (e.g., a database). To the best of our knowledge no one has analyzed the government's legal need, the user's civil rights and protection mechanisms, or the cost/efficiency justifications of escrow encryption as they pertain to various applications like long term storage channels. The access-control key associated with a storage channel should have a LEAF-like element which should enable wiretapping, and the granularity

of such keys depends on the definition of a proper "storage context" for wire-tapping. Finally, multicast and broadcast channels (rather than point-to-point) are another open issue.

7 Conclusions

We analyzed basic requirements of a sound key escrow system. Such a system is more complicated in parties, components, goals and exposures than traditional systems. We presented attacks and other basic concerns that apply to Clipper and recent designs, thus increasing our understanding of the nature of such systems. We believe that we are not exhaustive and that further analysis and technical discussions are needed.

Acknowledgments

Thanks to Chris Carrol for his help with the actual attack experiment. Helpful discussions on various aspects of key escrow, Clipper and this paper with Matt Blaze, Dorothy Denning, Yvo Desmedt, Matt Franklin, Michael Froomkin, Amir Herzberg, Paul Karger, David Kravitz, Hugo Krawczyk and Ron Rivest are acknowledged.

References

1. R. Anderson, *Why Cryptography Fails*, In the Proceedings of The 1st ACM Conference on Computer and Communications Security, Nov. 1993, 215–227.
2. D. M. Balenson, C. M. Ellison, S. B. Lipner and S. T. Walker *A New Approach to Software Key Escrow Encryption*, Trusted Information Systems, Inc., (also in [21]).
3. T. Beth, Zur Diskussion Gestellt, *Informatik-Spektrum* 13 (4), pp. 204-215, 1990. (An initial suggestion for Key Escrow System and Agent– a public presentation to the German Government, In German).
4. T. Beth, H.-J. Knobloch, M. Otten, G.J. Simmons and P.Wichmann, Towards Acceptable Key Escrow Systems, In the Proceedings of The 2nd ACM Conference on Computer and Communications Security, November 1994 51–58.
5. M. Blaze, *Protocol failure in the Escrowed Encryption Standard*, In the Proceedings of The 2nd ACM Conference on Computer and Communications Security, November 1994, 59–67. (also in [21]).
6. E.F. Brickel,et al. Interim Review: The SKIPJACK Algorithm, July 93. (Also in [21]).
7. D.E. Denning et al., *To Tap or Not To Tap*, CACM 93.
8. D. E. Denning and M. Smid, Key Escrowing Now, IEEE Communications Magazine, Sep. 1994, pp. 54-68.
9. Department of Justice, Letter dated Feb. 4, 1994
10. A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, *How to Share a Function Securely*, ACM STOC 94. (Initial version May 92: FOCS 92 submission).

11. Y. Desmedt, Securing Traceability of Ciphertexts: Towards a Secure Software Key Escrow Systems, Eurocrypt 95.
12. Y. Desmedt, Y. Frankel, and M. Yung, A scientific statement on the Clipper Chip technology and alternatives, (a letter to NIST as an answer to a request for comments on key escrow technology).
13. Y. Desmedt and Y. Frankel, *Threshold cryptosystems*, In G. Brassard, editor, *Advances in Cryptology, Proc. of Crypto '89 (Lecture Notes in Computer Science 435)*, pages 307–315. Springer-Verlag, 1990.
14. Y. Desmedt and J.-J. Quisquater, *Public-key systems based on the difficulty of tampering (Is there a difference between DES and RSA?)*, Advances in Cryptology– Proc. Crypto '86, Springer-Verlag LNCS 263, 1987, 111–117.
15. W. Diffie and M. Hellman, *New Directions in Cryptography* , IEEE Trans. on Information Theory 22 (6), 1976, pp. 644-654.
16. T. El Gamal, *A Public key cryptosystem and a signature scheme based on discrete logarithm*, IEEE Trans. on Information Theory 31, 465-472, 1985.
17. The FBI, Law Enforcement REQUIREMENTS for the Surveillance of Electronic Communications, June 1994. (Prepared by the Federal Bureau of Investigations (FBI) in cooperation with federal, state, and local law enforcement members of the National Technical Investigation Association).
18. FIPS PUB 185, *Escrowed Encryption Standard* February 1994. (Dept. of Commerce).
19. Y. Frankel and M. Yung, *Designs of escrow encryption systems: models, methodologies and technologies*, (Available from the authors).
20. Y. Frankel and M. Yung, Preliminary version of current paper, originally submitted to IEEE Security and Privacy 95 (Oakland), Nov. 94.
21. *Building in Big Brothers*: the cryptographic policy debate, ed. L.J. Hoffman, Springer Verlag, 1995.
22. J. Kilian and F.T. Leighton, *Failsafe Key Escrow Systems*, Crypto 95.
23. D. Kravitz, *Deficiencies of Software-based key escrow.* a letter.
24. S.M. Matyas, *Key Processing with Control Vectors*, Journal of Cryptology, 3 (2), pp 113-136, 1991
25. S. Micali, *Fair public-key cryptosystems*, Crypto '92 (also in [21]).
26. The Mosaic program office, *Mosaic: Cryptographic intertrace programmers guide for the Tessera crypto card*, Draft Revision P1.4.
27. MYK-78T Encryption/Decryption VLSI, Mykotronx Inc.
28. M. Naor and M. Yung, *Public-key cryptosystem provably secure against chosen ciphertext attack*, Proc. of the 22nd Annual Symposium on the Theory of Computing, 1990, pp. 427–437.
29. T. P. Pedersen, Distributed Provers with Applications to Undeniable Signature, Eurocrypt '91. 1991.
30. R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signature and Public Key Cryptosystems*, Comm. of ACM, 21 (1978), pp 120-126.
31. G. Simmons, *The Subliminal Channel and Digital Signature*, Eurocrypt 84, 51–67.
32. The White House Press Release Regarding the Clipper, the White House – office of the press secretary, April 16, 93.