University of Technology Chemnitz-Zwickau

Department of Computer Science

Theoretical Computer Science and Information Security

Technical Report

TR-95-4-F

A new key escrow system with active investigator

Patrick Horster · Markus Michels · Holger Petersen

April 1995

Straße der Nationen 62 Phone: (+49) 371-531-1479

D-09111 Chemnitz, Germany Fax: (+49) 371-531-1810

A new key escrow system with active investigator[1]

Patrick Horster · Markus Michels · Holger Petersen

Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau
Straße der Nationen 62, D-09111 Chemnitz, Germany
E-mail: {pho,mmi,hpe}@informatik.tu-chemnitz.de

April 18, 1995

Abstract

During the last two years there has been a stimulating discussion about escrow technologies. In this paper we give a brief review of two different concepts: The hardware based "Clipper" approach and a software based "escrowed key exchange" approach with active investigator. We point out the main weaknesses in both approaches and present a new software based escrowed key exchange protocol with active investigator that copes with all cryptographic weaknesses.

1. Introduction

The use of encryption in devices of every-day use (e.g. telephones) in order to prevent eavesdropping raises the problem that the government can't extract information by court-ordered line-tapping. Thus criminals can use such secured devices without any fear to be tapped. In contrast, the use of weak encryption systems threats the privacy of the users.

One suggestion to solve this conflict is a *key escrow system* which provides cryptographic protection to unclassified, sensitive data while allowing for government access to the data when lawfully authorized. In a key escrow system we distinguish several participants: The *escrow agency* holds the secret keys of all *users*. It can be divided into many *escrow agents* who hold shares of the secret keys. To guarantee the privacy of non-criminal users at least one escrow agent must be trustworthy. An *investigator* who is authorized by a *court* can tap the traffic of suspicious users. A *passive investigator* has to transfer this taped traffic to the escrow agency who checks the authorization and decrypts the messages for him. An *active investigator* obtains secret parameters from the escrow agency such that he can decrypt the traffic himself.

However, the adversaries of this concept fear that it helps to establish a totalitarian society, where every action of anybody can be monitored and supervised by a big brother, see [Orwe48] for details. Secret keys of the users might be leaked because of improper storage or dishonest key escrow agents. Furthermore, *if* the concept is once established then it will survive, even if weaknesses might become obvious. As a pre-encryption by criminals can't be avoided and the

[1] appeared in Proc. Securicom'95, Paris, (1995), pp. 15 – 28.

existence of a pre-encryption can only be recognized if the authorized investigator taps the line, the injunction of the pre-encryption and thus the restriction of private use of cryptographic methods might be enforced. As a subliminal channel, that is a private channel between two user whose existence is hidden, can be established by criminals every restriction of the private use of cryptographic methods is nonsense – according to the adversaries (e.g. see [MoPS94] for the establishment of such a channel during telephone calls using ISDN).

In contrary, the intercessors of this concept emphasize the need of court-ordered line-tapping to convict criminals, which would be more difficult without having this possibility. Although they have to admit that pre-encryption can't be avoided, they stress that it also helps the investigator to mention that a suspicious person pre-encrypts his messages. Additionally, pre-encryption or establishing a subliminal channel requires a certain amount of knowledge and costs. Therefore it won't be used at every time by every criminal. As nowadays most data is sent unencrypted over insecure channels the establishment of an escrow-system protected channel is a progress for the privacy of the users.

Another level of the discussion is given by the requirements which an acceptable key escrow system should satisfy. We are of the opinion that *if* a key escrow system should be established then the investigator should only be able to do exactly what he is authorized by the court-order. It must further be analyzed which proposals fulfill the given requirements. We discuss our requirements in section 3 in detail.

The debate about escrow systems intensified as the U.S.Government proposed the Escrowed Encryption Standard (EES) in 1993. However, as the EES contains several weaknesses from a cryptographic point of view, there has been a stimulating discussion about the new standard. Recently there has been proposed a software based approach with active investigator, which copes with some of the weaknesses in the Clipper proposal, but invokes new weaknesses, which we point out in this paper. To cope with these weaknesses, we introduce a new software based approach with active investigator, which keeps the advantages of the software based solution. These are namely the possibility of a cheap realization instead of using a tamper-proof chip, the publically known algorithm and the concept that the escrow is embedded into the key-exchange phase.

We review the Escrowed Encryption Standard in section 2, formulate some requirements and possible solutions in section 3 and review the escrowed key exchange protocol in section 4. We point out the main weaknesses of this approach in section 5 and present our new escrowed key exchange system in section 6, which fits the requirements and avoids all security flaws.

2. The Escrowed Encryption Standard

On April 16, 1993, the Government of the United States of America announced a proposed new symmetric key encryption system, called the Escrowed Encryption Standard (ESS), for protection of sensitive data [Mark93, NIS194]. The heart of this proposal is the tamper-proof encryption chip called the "Clipper Chip" together with shared key approach for key escrow [Denn93, Ruep94]. The Clipper Chip contains a classified single key 64 bit block encryption algorithm called "Skipjack". The algorithm uses 80 bit keys and has 32 rounds of scrambling. Each chip consists of the following components:

1. a family key F, that is common to all chips,
2. a Unit ID, that identifies each Clipper Chip,
3. a secret key U, that unlocks all messages encrypted with the chip.

Once a session key K is established using any asymmetric key exchange protocol, the Clipper Chip is used to encrypt the message stream m. The law enforcement field (LEAF) has the following structure:

$$LEAF = E_F(E_U(K)||UnitID||checksum).$$

The exchange of the LEAF is enforced by the Clipper Chip by decrypting the ciphertext only after the receipt of a valid LEAF. The detailed description of chip programming and escrow is given in [Denn93]. The algorithm is briefly given in figure 1. The authorization procedures for release of encryption key components to the active investigator by the escrow agents have been announced in [Reno94].

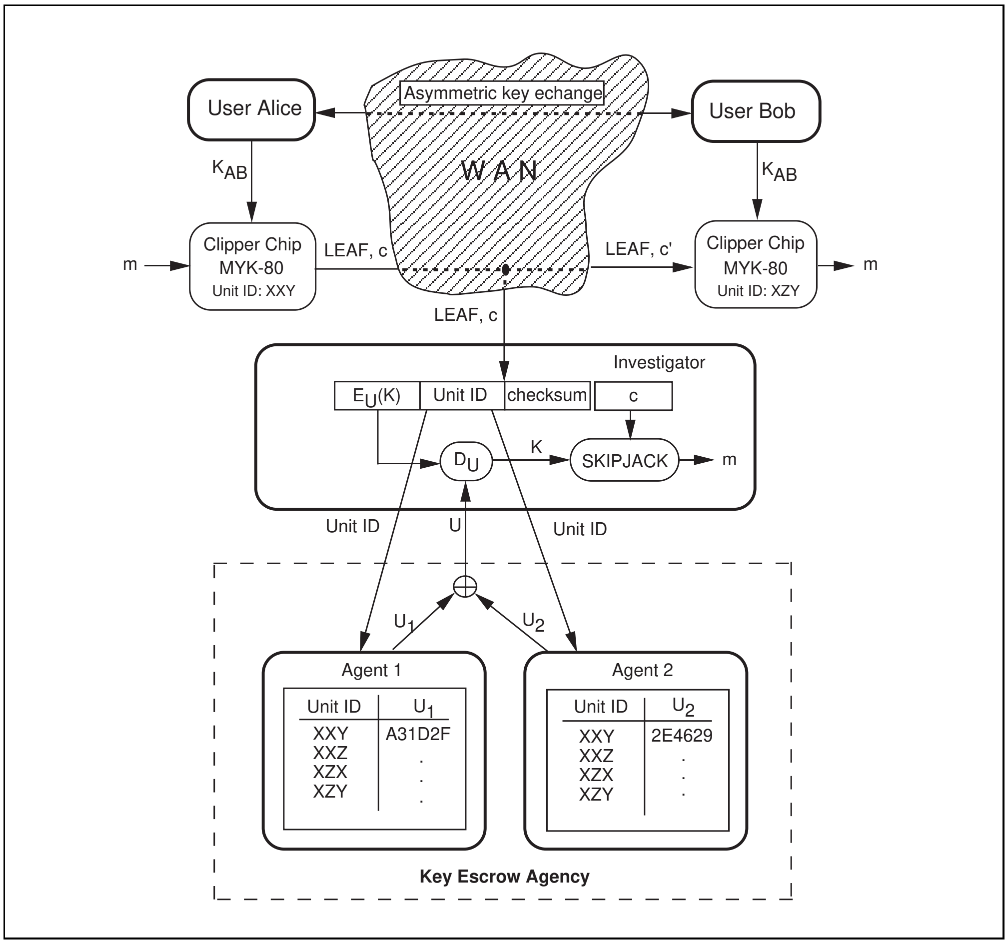

Figure 1: The Escrowed Encryption Approach

Weaknesses of the Clipper approach

There has been a long stimulating discussion about the EES, which is summarized in [LKBC94, NIS194]. The main shortcomings of the EES from a cryptographic point of view are:

- The "Skipjack" algorithm is classified and was thus not object of public review and criticism.
- Once interception is allowed, the active investigator can intercept all further messages and also decrypt the monitored older messages, as they obtain the unit key U from the escrow agents.
- Criminals using Clipper, can use their own (secret) super-encryption/decryption algorithms together with the chip, such that the active investigator can not decrypt the messages.
- Blaze has shown, how the Clipper Chip can be used without generating a valid LEAF, such that the Escrow Agency can't decrypt the ciphertext encrypted with the Skipjack algorithm [Blaz94], although the described method is not yet applicable for real-time communications [NIS294].
- Tamperproofness of the Clipper Chip is essential because otherwise the algorithm might be used with incorrect chosen LEAF [Desm95].

3. Requirements and solutions

To overcome the shortcoming in the Clipper approach that once the interception is allowed by a lawyer the authorities can intercept every further message and also decrypt the monitored older messages, we formulate one requirement which is a slight modification of statement 5 in [BKOS94].

Requirement: If wiretaps are warranted for a precisely defined time interval T, which can be located either in the past, the present or future, authorization by the escrow agency to the investigator must be limited so that exactly this time interval is interceptable.

There have been several proposals to cope with this requirement:

1. One solution, due to Diffie, is that the session key K_{AB} is splitted and that the LEAF contains the encrypted values $E_{U_1}(K_{AB,1})$ and $E_{U_2}(K_{AB,2})$ instead of $E_U(K_{AB})$. Then the escrow agency reveals the session key K instead of the unit key U. This solution has several drawbacks, as for example the key escrow agents must be involved in every decryption, such that it is not recommended [BKOS94].

2. Another solution is that the escrow agents could provide a (tamper proof) hardware module that performs the decryption function and doesn't reveal the unit key to the escrow agents [BKOS94].

   Other hardware based solutions exist [BeOt94], but tamper resistance is essential here.

3. There are three software based approaches so far [BeOt94, BKOS94, Desm95]. The solution due to Walker and Balenson uses an active investigator. However, the above requirement is not satisfied, as once the interception is allowed, the active investigator can intercept all further messages and also decrypt the monitored older messages, as he obtains the unit secret key from the escrow agents. The solution due to Desmedt uses a passive investigator, who only taps the encrypted traffic. As decryption of *every* session key is done by the escrow agents, this might lead to a bottleneck in communication and availability of the agents. The third software based solution uses a modified Diffie-Hellman key exchange protocol [DiHe76, Guen89] including an active investigator [BKOS94]. We give a more detailed description in the next section.

4. Our software based solution is based on an authentic key exchange protocol including an active investigator. It allows decryption of the message in a precisely defined time interval T without active participation of the escrow agents. The solution is described in detail in section 6.

4. The escrowed key exchange protocol

The main idea of this approach is that the investigator has to participate actively in the key exchange protocol to escrow the session key. As he doesn't get the user's secret key for the computation of the session key but only a related key, he can not compute any previously exchanged session keys using old traffic.

In the following we describe the escrowed key exchange protocol given in [BeOt94, BKOS94]. The message encryption can be done using any symmetric cryptosystem, e.g. the Skipjack algorithm.

Initialization

The trusted authority chooses primes p, q and q' with $q = 2q' + 1, q|(p-1)$, a generator α of a multiplicative subgroup of order q over $\mathbb{Z}_p$ and a generator β in $\mathbb{Z}_q$. These parameters are authentically known to all users.

Data for Alice's encryption device

User Alice chooses a secret key $x_A \in_R \mathbb{Z}_q$ and computes her public key $y_A := \alpha^{x_A} \pmod p$. Furthermore she chooses another secret value $t_A \in_R \mathbb{Z}_{q-1}$ for tapping.

Data for the escrow agency

The escrow agency (agents) possesses $\beta^{t_A} \pmod p$ and x_A for the user Alice.

Key exchange without interception

The key exchange consists of the exchange of two Diffie-Hellman halfkeys which are combined with an operation "$\circ$" (e.g. addition, multiplication, XOR) to get the session key. The protocol is the following:

User Alice		Network		User Bob
		$m_A, m_B \in_R \mathbb{Z}_q$		
		$z_B := \beta^{m_B} \pmod q$	$\longrightarrow$	$z_B \neq 0, \pm 1$
$z_A \neq 0, \pm 1$	$\longleftarrow$	$z_A := \beta^{m_A} \pmod q$		
$k_A := z_A^{t_A} \pmod q$				$k_B := z_B^{t_B} \pmod q$
$r_A := \alpha^{k_A} \pmod p$				$r_B := \alpha^{k_B} \pmod p$
r_B		$\longleftrightarrow$		r_A
$K_{1,A} := r_B^{x_A} \pmod p$				$K_{1,B} := y_A^{k_B} \pmod p$
$K_{2,A} := y_B^{k_A} \pmod p$				$K_{2,B} := r_A^{x_B} \pmod p$
$K_{A,B} := K_{1,A} \circ K_{2,A}$				$K_{A,B} := K_{1,B} \circ K_{2,B}$

The scheme is correct because the following congruence holds:

$$K_{A,B} := K_{1,A} \circ K_{2,A} = r_B^{x_A} \circ y_B^{k_A} \equiv \alpha^{k_B x_A} \circ \alpha^{k_A x_B} \equiv y_A^{k_B} \circ r_A^{x_B} = K_{1,B} \circ K_{2,B} =: K_{A,B} \pmod p.$$

Key exchange with interception

In the case of legal wiretapping, the investigator gets x_A, $\gamma := \beta^{u_A} \pmod p$, $u_A \in_R \mathbb{Z}_q$ and $\gamma^{t_A} \pmod p$ from the escrow agents. He sends $z_A := \gamma^{m_A} \pmod p$ to Alice, with randomly chosen m_A. Alice and Bob follow the protocol described above. The investigator computes $k_A := (\gamma^{t_A})^{m_A} \pmod p$ and the halfkey $K_{2,A} := y_B^{k_A} \pmod p$. As he knows Alice secret key x_A he can compute $K_{1,A} := r_B^{x_A} \pmod p$ and therefore $K_{A,B}$. The resulting protocol is given in figure 2.

5. Cryptanalysis of the escrowed key exchange protocol

There are five possible attackers: The users Alice and Bob, the investigator, the network and an outsider. The security against an outsider and network attack has been described in [BKOS94]. Here we focus on attacks by Alice, Bob and the investigator.

First weakness

An important possibility of cheating by *Alice* and *Bob* has already been mentioned in the security considerations of [BKOS94]. Alice can choose the parameter k_A randomly instead of following the protocol. After she gets z_A from the network she either chooses k_A randomly or computes $k_A := z_A^{u_A} \pmod p$ with a random value $u_A \in_R \mathbb{Z}_q$. Then she follows the protocol. As the solution is hardware independent, we don't see how to prevent this attack. The serious flaw here is that a session key can be established by Alice and Bob (even if Bob is innocent) while the network doesn't notice that Alice cheats. The investigator is able to identify the cheating party later, when he either obtains a different key $\tilde{r}_A$ from his computation or can't read any message. But what can he do in this case ? If he tells Alice that she should stop cheating then she is warned and won't send any interesting messages over the network. Thus it is essential that deviation from the protocol can always be noticed by the network. A hardware based solution has already been proposed in [BKOS94], but then the advantages of the software based solution are lost.

Second weakness

The *investigator* can try to wiretap the channel although he is not allowed to do so. Once he knows the parameters he can compute *every* session key he wants to know. The fifth statement in [BKOS94] states that authentication must be limited so that exactly a precisely defined

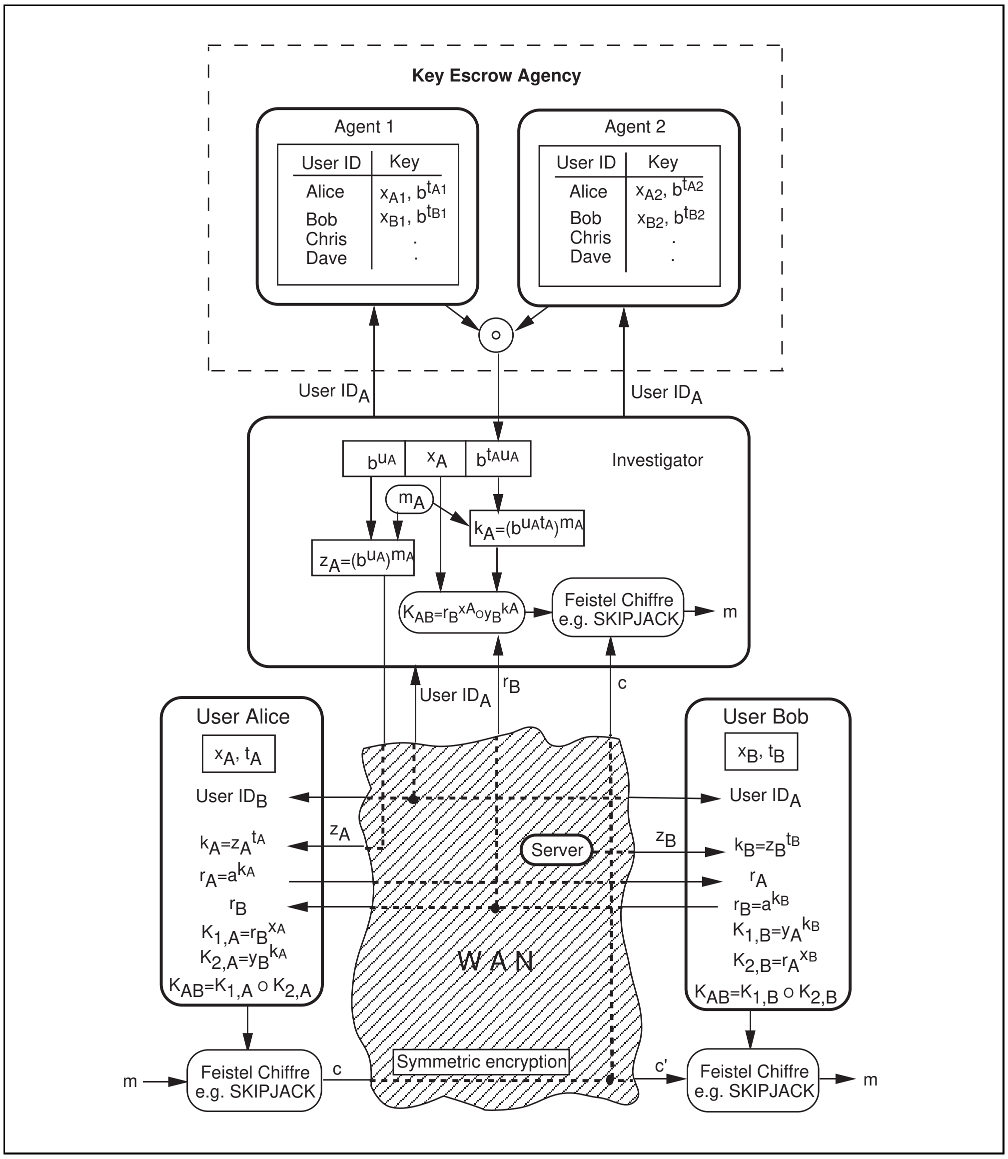

Figure 2: The escrowed key exchange with interception

time interval T is interceptable. This requirement is not achieved, because after a successful interception every further session key can be computed by the investigator without any time limit.

Third weakness

The third weakness concerns the case, in which the *investigator* has a suspicion and wants to intercept a telephone line, but can't get a court order for interception in time. In this case all previously made communications are lost for the investigator, as he can only read those messages, when he participated actively in the protocol. E.g. if a criminal has been identified it might be helpful for the police to decrypt earlier communications of this person. In this case it is likely that the police was not actively participating in the protocol at earlier time, when they didn't have their suspicion yet. In the following we suggest a protocol by which the investigator can still read the messages later, after he got the court order, by decrypting the taped communications.

6. The new escrowed key exchange protocol

Now we present the solution to cope with the three weaknesses. The idea is, that the random parameters k_A and r_A which can't be verified by the network in the protocol above, are used in an ElGamal like signature scheme [HoMP94]. Thereby they are related and can't be substituted randomly.

To cope with the problem that interception is possible only during a specific time interval $I_t = [t : t + temp]$, Alice possesses different secret and public key for each interval. This has the advantage, that after the time period $temp$, the investigator doesn't know the new secret key and thus can't decrypt the session key without a new permission of the escrow agency.

The protocol can be divided into two separate steps: The first step is the generation of distinct certified keypairs for each time interval and the second step is the authentic key exchange using these certified keys.

6.1 Generation of certified keypairs

Suppose that the time period can be divided into n time intervals I_t of duration $temp$. For example, if the user refreshes his keypairs once a year and we have time intervals of one week, then the total number of intervals is $n = 52$ and therefore the user needs 52 certified keypairs.

A simple solution to generate certified keypairs $(x_{A,i}, y_{A,i})$, $1 \leq i \leq n$, with $y_{A,i} := \alpha^{x_{A,i}} \pmod p$ for a user Alice is, that each escrow agent j, $1 \leq j \leq l$, gets a share $x_{A,i,j}$ of the secret key $x_{A,i}$ from Alice such that the sum of all shares is her secret key, e.g. $x_{A,i} := \sum_{j=1}^{l} x_{A,i,j} \pmod q$. Then each agent computes the corresponding public key of his share $y_{A,i,j} := \alpha^{x_{A,i,j}} \pmod p$ and the product $y_{A,i} := \prod_{j=1}^{l} y_{A,i,j} \pmod p$ including the time stamp is certified. This certificate can be verified by everyone and indicates, that the escrow agency knows the secret key for tapping purpose.

To reduce the memory requirements for the escrow agency it is recommended that a share $x_{A,i,j}$ of the secret key $x_{A,i}$ is computed as $x_{A,i,j} := E_{SK_{A,j}}(x_{A,i-1,j})$ using a secret master key $SK_{A,j}$ known to the user and the escrow agent j, a share $x_{A,0,j}$ and a symmetric cryptosystem with encryption function E. Then the escrow agent j has to store only two parameters $x_{A,0,j}$ and $SK_{A,j}$ for Alice. As each $SK_{A,j}$ is never revealed to the network or the investigator there is no fear that the investigator can compute the secret key for the time interval I_t if he knows any other secret keys for other time intervals. The drawback on this solution is that the network can't compute the corresponding public keys $y_{A,i}$, $1 \leq i \leq n$, for himself using a certified public key $y_{A,0}$. Thus the certified public keys must be provided to him by the escrow agency or the user.

It would be desirable that additionally the network can compute the corresponding public keys $y_{A,i}, 1 \leq i \leq n$, using a function F_A with the certified public key $y_{A,0}$ as argument. However, it remains an open problem how this function can be chosen.

6.2 The Escrowed Key Exchange Protocol

Initialization

The trusted authority chooses primes p, q and q with $q|(p-1)$, a generator α of a multiplicative subgroup of order q over $\mathbb{Z}_p$ and a collision free hash function h. These values are authentically known to all users.

Data for Alice's encryption device

User Alice chooses n secret keys $x_{A,i} \in_R \mathbb{Z}_q$ and computes the related public keys $y_{A,i} := \alpha^{x_{A,i}} \pmod p$. These keys are certified by the escrow agency, including a time stamp indicating the interval I_t during which they remain valid.

Data for the escrow agency

The escrow agency (agents) possesses all n secret keys $x_{A,i}$ from the user Alice.

Key exchange without interception

During the time interval I_t in which Alice uses her keypair $(x_{A,t}, y_{A,t})$ and Bob his keypair $(x_{B,t}, y_{B,t})$ the following protocol is used, where $\oplus$ denotes bitwise XOR:

User Alice		Network		User Bob
m_A	$\longleftarrow$	$y_{A,t}, y_{B,t}$ authentic $m_A, m_B \in_R \mathbb{Z}_q$	$\longrightarrow$	m_B
$k_A \in_R \mathbb{Z}_q$				$k_B \in_R \mathbb{Z}_q$
$r_A := \alpha^{k_A} \pmod p$				$r_B := \alpha^{k_B} \pmod p$
$s_A := x_{A,t} \cdot (r_A \oplus m_A) + k_A$	$\longrightarrow$	r_A, s_A		
		$\alpha^{s_A} \equiv y_{A,t}^{r_A \oplus m_A} r_A$?	$\longrightarrow$	r_A
		r_B, s_B	$\longleftarrow$	$s_B := x_{B,t} \cdot (r_B \oplus m_B) + k_B$
r_B	$\longleftarrow$	$\alpha^{s_B} \equiv y_{B,t}^{r_B \oplus m_B} r_B$?		
$K_{1,A} := r_B^{x_{A,t}} \pmod p$				$K_{1,B} := y_{A,t}^{k_B} \pmod p$
$K_{2,A} := y_{B,t}^{k_A} \pmod p$				$K_{2,B} := r_A^{x_{B,t}} \pmod p$
$K_{A,B} := h(K_{1,A} \circ K_{2,A})$				$K_{A,B} := h(K_{1,B} \circ K_{2,B})$

The scheme is correct as the following congruence holds:

$$\begin{aligned} K_{A,B} &:= h(K_{1,A} \circ K_{2,A}) = h(r_B^{x_{A,t}} \circ y_{B,t}^{k_A}) \equiv h(\alpha^{k_B x_{A,t}} \circ \alpha^{k_A x_{B,t}}) \\ &\equiv h(y_{A,t}^{k_B} \circ r_A^{x_{B,t}}) = h(K_{1,B} \circ K_{2,B}) =: K_{A,B} \pmod p. \end{aligned}$$

The function h is chosen as a collision free hash function. This is necessary as the underlying key distribution system is vulnerable to triangle attacks if the session key $K_{A,B}$ is used [Burm94]. A dishonest user Chris can compute a session key of a former conversation of Alice and Bob, if he leads Alice and Bob to believe that he wants to agree a session key with each of them. Using the knowledge he gets from these protocols (where he behaves dishonest) *and* additional knowledge which Alice and Bob gives him voluntarily he can compute the old session key of them. This attack is primarily theoretical as neither Alice nor Bob give Chris this additional information. However, by the use of the hash function h this attack can be prevented [Burm94].

Key exchange with interception

Now we consider the case, when an (authorized) investigator participates in the protocol during the time interval I_t. He gets the corresponding secret key $x_{A,t}$ from the escrow agency presenting his court order. Using this key he computes $K_{1,A} := r_B^{x_{A,t}}$ and the secret value k_A by:

$$k_A = s_A - x_{A,t} \cdot (r_A \oplus m_A) \pmod q.$$

Using this value he obtains the session key

$$K_{A,B} := h(K_{1,A} \circ y_B^{k_A} \pmod p))$$

which he needs to decrypt the ciphertext c. The protocol is given in figure 3.

6.3 Security analysis

We show, how the three weaknesses are avoided in our protocol.

1. **First weakness:** As the network knows the public key $y_{A,t}$ for each time interval I_t, Alice can't replace $x_{A,t}$ by any other value, which the escrow agency is not able to compute. Thus the investigator can always compute k_A if he is aware of $x_{A,t}$.

   Alice can't replace k_A by a random value, as k_A and r_A are used for signature verification by the network. The network only forwards those values r_A for which the verification equation is true, and thus r_A has to be computed correctly.

2. **Second weakness:** As there are different (unrelated) keypairs $(x_{A,t}, y_{A,t})$ for every time interval I_t, the investigator can only decrypt the communication during that time interval.

3. **Third weakness:** As the computations can be done by using the old traffic $\{ID_A, (m_A, r_A, s_A), c\}$ which can be taped by the network, the investigator can always decrypt old ciphertext if he gets the old secret key from the escrow agency.

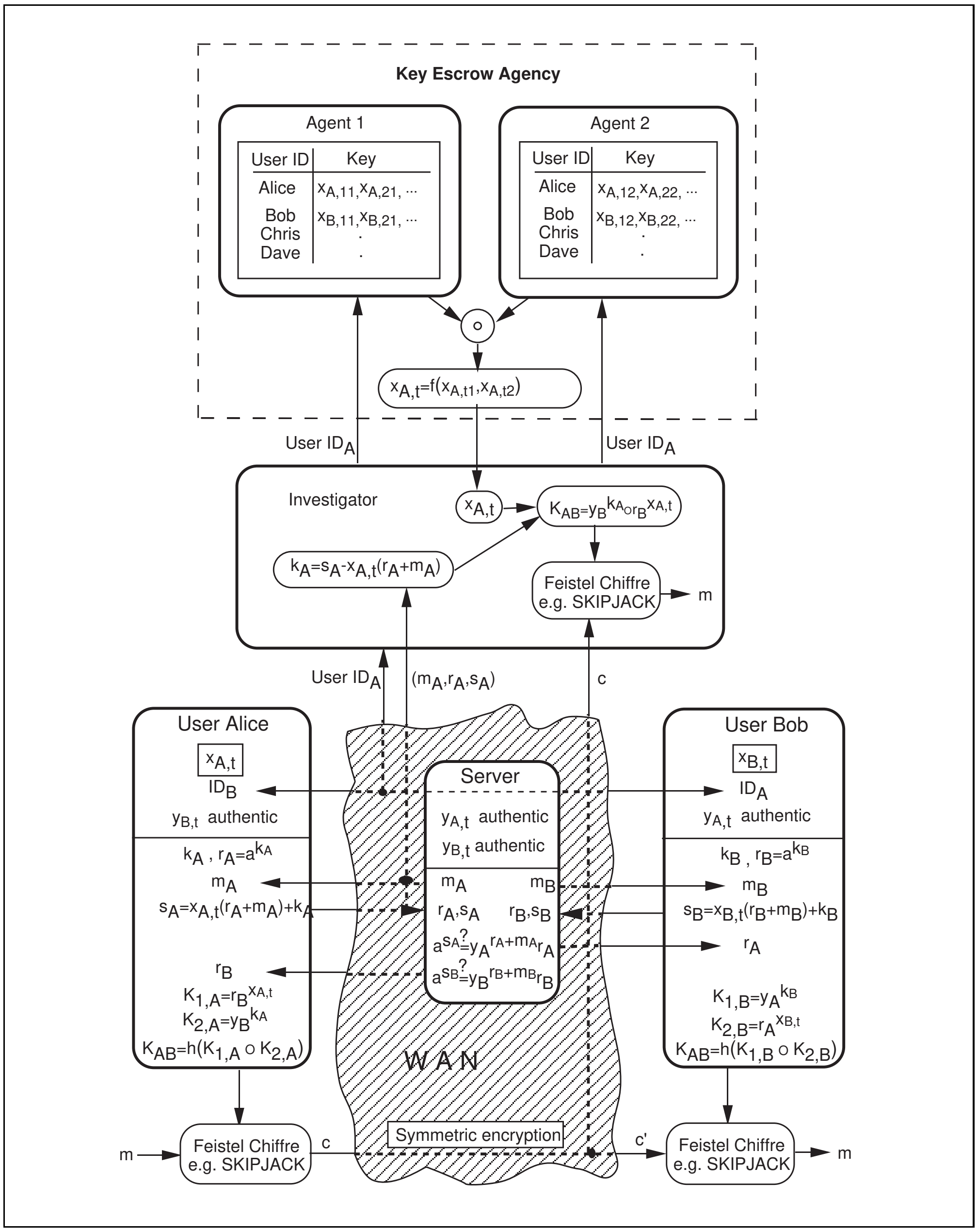

Figure 3: The new escrowed key exchange with interception

4. **Attacks by the network:** To avoid that the network can choose the messages m_A and m_B adaptively to get any information about the secret keys $x_{A,t}$ or $x_{B,t}$, we can slightly modify the protocol as follows: The user can either choose a second random message $\tilde{m}_A$, sign $h(m_A, \tilde{m}_A)$ and submit $\tilde{m}_A$ together with the signature or he can exchange a Diffie-Hellman key with the network at the beginning, which is then used as random message. The second solution additionally prevents outsiders from eavesdropping the signed message.

In a simplified version of the protocol the parameters (k_A, r_A) can be used as a certified keypair. Then the signature generation and verification is not necessary. This reduces the computational effort for the network. The disadvantage with this modification is that two parties either have to use the same session key for the whole time interval or get new certified keypairs (k_A, r_A) for each session while they can have arbitrarily many different ones in our original protocol using only one certified keypair $(x_{A,t}, y_{A,t})$. Thus either the security decreases or the computational effort increases. As a result, this modification is not recommended.

7. Conclusion

We reviewed the Escrow Encryption Standard and the escrowed key exchange protocol with active investigator presented recently. Then we pointed out the main weaknesses in both approaches and gave an overview about proposed solutions, which have not always successfully achieving the requirements. Finally we presented a new software based escrowed key exchange protocol with active investigator that copes with all cryptographic weaknesses discussed before. Our approach is a step towards designing key escrow protocols which achieve only those requirements which are absolutely necessary for the escrow agency. The limit of escrowed key exchange protocols is that it can't be avoided that criminals use other channels to compute a common key or they encrypt the session key again, e.g. using a secret hash function.

To prevent the user's privacy if the shares of his secret key are leaked by improper use or storage it might be wise to share just a part of the secret key while keeping a small part (e.g. 30–40 bit) totally secret by the user. In the case of tapping, this part can be computed using brutal force methods by the investigator. As a result, the user is at least protected by a weak encryption system, even if all shares of his secret key are leaked, which are kept by the escrow agents.

8. Acknowledgements

The authors would like to thank the participants of the platform discussion about the crypto conflict at the VIS'95 (Reliable information technical systems) conference in Rostock, Germany. Furthermore the authors are very much obliged to an anonymous reviewer, whose name shouldn't be escrowed, for helpful discussion and several comments.

References

[BKOS94] T.Beth, H.-J.Knobloch, M.Otten, G.Simmons, P.Wichmann, "Towards Acceptable Key Escrow Systems", Proc. 2nd ACM Conference on Computer and Communications Security, Fairfax, Nov. 2–4, (1994), pp. 51–58.

[BeOt94] T.Beth, M.Otten, (ed.) "E.I.S.S.–Workshop on Escrowed Key Cryptography", E.I.S.S.-Report 94/7, University of Karlsruhe, June 22 – 24, (1994), 160 pages.

[Blaz94] M.Blaze, "Protocol Failure in the Escrowed Encryption Standard", Proc. 2nd ACM Conference on Computer and Communications Security, (1994), pp. 59–67.

[BDKM93] E.F.Brickell, D.E.Denning, S.T.Kent, D.P.Mahler, W.Tuchman, "SKIPJACK Review", Interim Report, July 28, (1993), 8 pages.

[Burm94] M.Burmester, "On the Risk of Opening Distributed Keys", Lecture Notes in Computer Science , Advances in Cryptology: Proc. Crypto '94, Berlin: Springer Verlag, (1994), pp. 308–317.

[Denn93] D.Denning, "The Clipper Chip: A Technical Summary", April 21, (1994), 3 pages.

[Desm95] Y.Desmedt, "Securing Traceability of Ciphertexts – Towards a Secure Software Key Escrow System", Advances in Cryptology: Proc. Eurocrypt '95, Lecture Notes in Computer Science, Berlin: Springer Verlag, (1995).

[DiHe76] W. Diffie, M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November, (1976), pp. 644–654.

[Guen89] C.G.Günther, "An identity based key exchange protocol", Lecture Notes in Computer Science 434, Advances in Cryptology: Proc. Eurocrypt '89, Berlin: Springer Verlag, (1990), pp. 29–37.

[HoMP94] P.Horster, M.Michels, H.Petersen, "Meta-ElGamal signature schemes", Proc. 2. ACM conference on Computer and Communications security, Fairfax, Virginia, 2.-4. Nov., (1994), pp. 96 – 107[1].

[LKBC94] S.Landau, S.Kent, C.Brooks, S.Charney, D.Denning, W.Diffie, A.Lauck, D.Miller, P.Neumann, D.Sobel, "Codes, Keys and Conflicts: Issues in U.S. Crypto Policy", Report of a Special Panel of the ACM U.S. Public Policy Committee (USACM), June, (1994), 74 pages.

[Mark93] J.Markoff, "Communications plan to balance government access with privacy", New York Times, April 16, (1993).

[MoPS94] S.Möller, A.Pfitzmann, I.Stierand, "Rechnergestützte Steganographie: Wie sie funktioniert und warum folglich jede Reglementierung von Verschlüsselung unsinnig ist", Datenschutz und Datensicherheit, 18/6, (1994), pp. 318–326.

[NIS194] National Institute for Standards and Technology, "Escrowed Encryption Standard", Federal Information Processing Standards Publication 185, U.S. Dept. of Commerce, February 9, (1994).

[NIS294] National Institute for Standards and Technology, "Technical Fact Sheet on Blaze Report and Key Escrow Encryption", June 15, (1994), 2 pages.

[Orwe48] G.Orwell, "1984", (1948).

[Reno94] J.Reno, "Key Escrow Encryption Announcements", Department of Justice, U.S. Government, February 4, (1994).

[Ruep94] R.A.Rueppel, "Clipper – Der Krypto-Konflikt am Beispiel der Amerikanischen ES-CROW Technologie", Datenschutz und Datensicherheit, 18/8, (1994), pp. 443–451.

[1] Available via anonymous FTP in the directory *ftp.tu-chemnitz.de/pub/Local/informatik/crypto* or via WWW under URL <http://www.tu-chemnitz.de/~hpe/techrep.html>