Attacks on the HKM / HFX Cryptosystem

Xuejia Lai and Rainer A. Rueppel

r[3] security engineering ag, Zurichstrasse 151, CH - 8607 Aathal, Switzerland
Tel: +41 1 934 5656, Fax: +41 1 934 56 57, Email: lai@r3.ch, rueppel@r3.ch

Abstract

The HKM / HFX cryptosystem is proposed for standardization at the ITU Telecommunication Standardization Sector Study Group 8. It is designed to provide authenticity and confidentiality of FAX messages at a commercial level of security. In addition, the HKM / HFX cryptosystem is designed for unrestricted export.

This paper contains the results of an analysis of the HKM / HFX cryptosystem. Eleven attacks and their complexities are described in full detail. The analytic results show that the security provided by the HKM / HFX cryptosystem is not high enough to meet the requirements for an international standard of the ITU, even with the additional feature of free exportability.

1 Introduction

The HKM / HFX cryptosystem is proposed for standardization at the ITU Telecommunication Standardization Sector Study Group 8. The goal of the system is to provide authenticity and confidentiality of FAX messages at a commercial level of security. In addition, the HKM / HFX cryptosystem is designed for unrestricted export.

This paper contains the results of an analysis of the HKM / HFX cryptosystem. We show eleven attacks on the system and the estimated complexities of these attacks. Section 2 is a description of the cryptosystem based on the original documents [1-4]. Section 3 is the summary of the analysis results where the we give a outline of the relationship between different attacks, the assumptions for each attack and the complexities. Details of the attacks are given in Section 4.

2 The HKM / HFX Cryptosystem

The system consists of two stages. In the first stage, called *Registration Mode* and shown in Fig.1, a common secret master key for communication from fax machine A to fax machine B is generated by A and sent to B. All the fax messages from A to B are sent in the second stage shown in Fig. 2, called *Automatic Mode*, in which a session key is established between A and B and it is used to protect the fax message.

2.1 Notations

Abbr.	Digits	Explanation
F_A	6	Last 6 digits of the fax-number of machine A
ID_A	48	Identity string of fax machine A.
$CRYP_A$	16	Unique cryptographic string of fax machine A. This string corresponds to an individual master key.
MP_{AB}	16	Mutual primitive from fax machine A to fax machine B. This string corresponds to a unidirectional master key for transmissions from A to B.
OT_{AB}	6	One-time key shared between A and B. Needs to be transferred by secure out-of-band methods.
TK_{AB}	16	Transfer key from A to B. This string is the mutual primitive MP_{AB} enciphered with the one-time key shared between A and B.
RCS_{AB}	16	Registered Crypt String, which is the mutual primitive MP_{AB} enciphered by machine B using (F_A,F_B,ID_B,$CRYP_B$) as the key. It is sent from B to A at the registration mode and stored at machine A . In automatic mode, it is sent from A to B and used by B to recover the session key SK.
SK	12	Session key chosen by the sending fax machine A at the beginning of a confidential transmission and transferred to the receiving machine B enciphered under the mutual primitive MP_{AB} and a random string RS.
RS	4	Random string chosen by the sending fax machine A as additional key input to the encipherment of the session key SK at the beginning of a confidential transmission. Transferred in plain to the receiving machine B.

2.2 Registration Mode

1. The sending machine A and receiving machine B exchange a 6-digit *secret one time key* **OT_{AB}** in a secure way outside the actual transmission.
2. A 16-digit *mutual primitive* **MP_{AB}** is generated at machine A from the identity string ID_A, the unique crypt string $CRYP_A$ and the fax numbers of both machines. MP_{AB} is in fact the unidirectional master key for transmissions from A to B, and remains unchanged during the life time of both machines.
3. The MP_{AB} is then encrypted to a 16-digit *transfer key* **TK_{AB}** at machine A by using the HKM algorithm under the one time key OT_{AB}.
4. A sends the TK_{AB} to machine B.

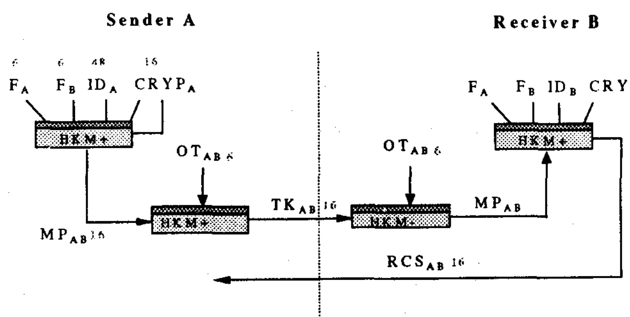

Figure 1. The registration mode of the HKM system where HKM+ denotes the encryption operation and HKM- denotes the decryption.

5. The receiving machine B decrypts TK_{AB} using the one time key OT_{AB} to get MP_{AB}.
6. B encrypts MP_{AB} to a 16-digit *registered crypt string* **RCS_{AB}** using $CRYP_B$, ID_B and the fax numbers of both machines.
7. B sends RCS_{AB} to A and A stores the RCS_{AB} together with machine B's fax number.

2.3 Automatic Mode

The confidential transmission of fax messages from machine A to machine B is carried out in the automatic mode.

1. A reproduces the MP_{AB}.
2. A generates a 12-digit *random session key* **SK**, encrypts SK to a 12-digit *encrypted session key* **ESK** using MP_{AB} and an additional 4-digit *random string* **RS**.
3. A sends the registered crypt string RCS_{AB}, the encrypted session key ESK and the random string RS to B.
4. B obtains MP_{AB} by deciphering RCS_{AB} and recovers SK by deciphering ESK.
5. All the fax messages transmitted from A to B in the session are encrypted at machine A and decrypted at machine B by using the HFX40 algorithm under the session key SK.

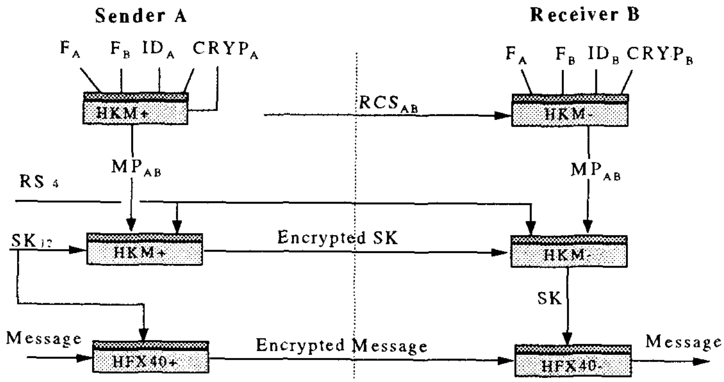

Figure 2. The automatic mode of the HKM system.

2.4 The HKM / HFX algorithms

A short description of the HKM / HFX algorithms is given in this section. In the HKM/HFX system, each fax machine is fabricated with 19 primes $p_0, p_1, \ldots, p_{17}, p_{18}$:

32603 32507 32183 32003 31847 31607 31583 31547 31259 31139

30803 30539 30467 30347 30323 29879 29759 29663

Each fax machine A is manufactured with a *unique 48-digit identity string* **ID_A**, and a *unique 16-digit crypt string*, **$CRYP_A$**.

2.4.1 The HKM algorithm

The HKM algorithm is used to generate and encrypt the key materials.

The basic HKM encryption system consists of the following parts:

- Key input processing, where the key input (which is 76 digits for computing the MP_{AB}, 6 digits for computing TK_{AB} and 20 digits for enciphering SK) is used to formulate 18 integers P[0],...,[8],B[0],...,B[8];

- Computing (decimal) keystream from the P[i],B[j].
- Encrypting plaintext string by digitwise addition mod 10 with the keystream.

Key input

↓

P[0] P[1] .... P[7] P[8] B[0] B[1] .... B[7] B[8]

↓

$$k[i] = \sum_{j=0}^{8} (P[j] * B[j]^{i+1} \bmod p_j) \bmod 10 \qquad i = 0,1,..$$

↓

Plaintext digits → ADD mod 10 → Ciphertext digits

2.4.2 The HFX 40 algorithm

The HFX 40 algorithm is used only to encrypt fax messages under a 12-digit (40-bit) session key SK. It has a similar structure to the HKM encryption algorithm. It consists of the following parts:

- Key input processing where the 12 digit input SK is used to select the moduli and to form integers P[0],P[1],P[2],B[0,B[1],B[2];
- Generation of binary keystream from the P[i],B[j];
- encryption plaintext by XOR

Key input (12 digits)

↓

P[0] P[1] P[2] B[0] B[1] B[2]

↓

$$k_i = (P[0]B[0]^{(i \bmod 1021)+1} \bmod p_0) \oplus (P[1]B[1]^{(i \bmod 1019)+1} \bmod p_1) \oplus (P[2]B[2]^{(i \bmod 1013)+1} \bmod p_2)$$

↓

Plaintext bits → ⊕ → Ciphertext bits

2.5 Generation of Mutual Primitive MP_{AB}

- The unique identity string ID_A of 48 digits is divided into a string of integers id[0],id[1],....,id[12] for which id[7] and id[8] are 2 digits and others are 4 digits integers.
- The unique crypt string $CRYP_A$ of 16 digits is divided into a string of three 4-digit integers and two 2-digit integers C[0],...,C[4].
- The fax-number of sending machine A, F_A, of 6 digits is split into two 3-digit integers $F_A[0]$ and $F_A[1]$.
- The fax-number of receiving machine B, F_B, of 6 digits is split into two 3-digit integers $F_B[0]$ and $F_B[1]$.

Unique ID_A string of 48 digits (sending machine)													$CRYP_A$ 16 digits				
4	4	4	4	4	4	4	2	2	4	4	4	4	4	4	4	2	2
id[0]	id[1]						id[7]	id[8]	id[9]			id[12]	C[0]	C[1]	C[2]	C[3]	C[4]
	+101	+202	+303	+404	+505	+606	+707	+808		+79	+2*79	+3*79	+4*79	+5*79	+6*79	+7*79	+8*79
$+F_A[0]$	$+F_A[1]$	$+F_B[0]$	$+F_B[1]$														
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]

- 18 integers P[i] and B[i], i=0,..9, are formulated as follows:

$P[0]=id[0] + F_A[0]$ $P[1]=id[1] + F_A[1] + 101$
$P[2]=id[0] + F_B[0] + 202$ $P[3]=id[1] + F_B[1] + 303$ $P[i]=id[i] + i\times101, \quad i = 4,\ldots,8$
$B[0]=id[9]$ $B[1]=id[10] + 79$ $B[2]=id[11] + 2\times79$ $B[3]=id[12] + 3\times79$
$B[4]=C[0] + 4\times79$ $B[5]=C[1] + 5\times79$ $B[6]=C[2] + 6\times79$ $B[7]=C[3]+7\times79$
$B[8]=C[4] + 8\times79$

- The Mutual Primitive MP_{AB} (which is the unidirectional master key for transmissions from machine A to machine B) is computed as follows:

$$MP_{AB}[i] = CRYP_A\,[i] + \sum_{j=0}^{8} \left(P[j] * B[j]^{i+1} \bmod p_j \right) \bmod 10 \qquad i= 0,1,\ldots15 \qquad (1)$$

2.6 Calculation of Transfer Key TK_{AB}

The transfer key TK_{AB} is computed by A and transmitted from A to B. It is the mutual primitive MP_{AB} enciphered with the one-time key shared between A and B.

- The 6-digit OT_{AB} is concatenated with itself to form a 64-digit input, which is then split into 14 4-digit and four 2-digit numbers, each is added by some integers as shown in the following table to form the numbers P[i] and B[j].

OT_{AB}	OT_{AB}		OT_{AB}	OT_{AB}		OT_{AB}		OT_{AB}		OT_{AB}	OT_{AB}		OT_{AB}	OT_{AB}		OT_{AB} (4)	
4-digit	4-digit	4-digit	4-digit	4-digit	4-digit	4-digit	2-digit	2-digit	4-digit	4-digit	4-digit	4-digit	4-digit	4-digit	4-digit	2-digit	2-digit
	+101	+202	+303	+404	+505	+606	+707	+808		+79	+2*79	+3*79	+4*79	+5*79	+6*79	+7*79	+8*79
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]

- The transfer key is obtained by

$$TK_{AB}[i] = MP_{AB}\,[i] + \sum_{j=0}^{8} \left(P[j] * B[j]^{i+1} \bmod p_j \right) \bmod 10 \qquad i= 0,1,\ldots15 \qquad (2)$$

2.7 Calculation of Registered Crypt String RCS_{AB}

At the receiving machine B:

- The unique identity string ID_B of 48 digits is divided into a string of integers id[0],id[1],....,id[12] for which id[7] and id[8] are 2 digits and others are 4 digits integers;
- The unique crypt string $CRYP_B$ of 16 digits is divided into a string of three 4-digit integers and two 2-digit integers;
- The fax-number of sending machine A, F_A, of 6 digits is split into two 3-digit integers $F_A[0]$ and $F_A[1]$;
- The fax-number of receiving machine B, F_B, of 6 digits is split into two 3-digit integers $F_B[0]$ and $F_B[1]$;

ID_B string 48 digits (receiving machine)													$CRYP_B$ 16 digits				
4	4	4	4	4	4	4	2	2	4	4	4	4	4	4	4	2	2
id[0]	id[1]						id[7]	id[8]	id[9]			id[12]					
	+101	+202	+303	+404	+505	+606	+707	+808		+79	+2*79	+3*79	+4*79	+5*79	+6*79	+7*79	+8*79
$+F_A[0]$	$+F_A[1]$	$+F_B[0]$	$+F_B[1]$														
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]

- The registered crypt string RCS_{AB} is obtained by enciphering the mutual primitive MP_{AB}

$$RCS_{AB}[i] = MP_{AB} + \sum_{j=0}^{8} \left(P[j] * B[j]^{i+1} \bmod p_j \right) \bmod 10 \qquad i=0,1,\ldots,15. \qquad (3)$$

2.8 Encryption of Session Key SK

- The mutual primitive MP_{AB} is repeated to form a 64-digit string which is then split into 14 4-digit and four 2-digit numbers.
- The 4-digit random string RS is split into two 2-digit integers S[0] and S[1].
- Integers P[i] and B[j] are formed from the above numbers as shown in the table

MP_{AB}				MP_{AB}					MP_{AB}				MP_{AB}				
$4_{0\text{-}3}$	$4_{4\text{-}7}$	$4_{8\text{-}11}$	$4_{12\text{-}}$	$4_{0\text{-}3}$	4	4	$2_{12\text{-}}$	$2_{14\text{-}}$	4	4	4	4	4	4	4	2	2
	+101	+202	+303	+404	+505	+606	+707	+808		+79	+2*79	+3*79	+4*79	+5*79	+6*79	+7*79	+8*79
+S[0]	+S[1]																
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]

- The 12-digit session key SK is encrypted to ESK by

$$ESK[i] = SK[i] + \sum_{j=0}^{8} \left(P[j] * B[j]^{i+1} \bmod p_j \right) \bmod 10 \qquad i=0,1,\ldots 11 \qquad (4)$$

2.9 Encryption of fax message with HFX 40 Algorithm

The HFX algorithm is similar to the HKM algorithms. It consists of the following parts:

1. The 12-digit session key SK is divided into four 3-digit integers, the first three integers (mod 19) are used to select the three modulo primes. That is,

$$p_0 <= p_{(SK(012) \bmod 19)} \quad p_1 <= p_{(SK(345) \bmod 19)} \quad p_2 <= p_{(SK(678) \bmod 19)}$$

where SK(012) stands for the integer determined by the first 3 digits, digit 0,1 and 2, of the session key SK. The 12-digit SK is then divided into six 2-digit integers, each is added by 1024 to form *the P[i], B[j].*

SK[01]	SK[23]	SK[45]	SK[67]	SK[89]	SK[ab]
+1024	+1024	+1024	+1024	+1024	+1024
P[0]	P[1]	P[2]	B[0]	B[1]	B[2]

2. Three binary strings, X of length 1021, Y of length 1019 and Z of length 1013 are obtained as

$$x[i] = \left(P[0] \times B[0]^{i+1} \bmod p_0 \right) \bmod 2 \qquad i=0,1,\ldots 1020$$
$$y[i] = \left(P[1] \times B[1]^{i+1} \bmod p_1 \right) \bmod 2 \qquad i=0,1,\ldots 1018$$
$$z[i] = \left(P[2] \times B[2]^{i+1} \bmod p_2 \right) \bmod 2 \qquad i=0,1,\ldots 1012$$

3. The binary fax message sequence $m_0, m_1, \ldots,$ is encrypted to the ciphertext sequence $c_0, c_1, \ldots,$ as

$$c_i = m_i \oplus x[i \bmod 1021] \oplus y[i \bmod 1019] \oplus z[i \bmod 1013] \quad i=0,1,\ldots$$

3 Outlines of the analytic results

The analysis has lead the following attacks on the HKM / HFX cryptosystem:

Attack 1. A ciphertext-only 'brute-force' attack which can break the HKM / HFX cryptosystem in $3 \cdot 10^6$ trial encryptions. „Break“ refers to the recovery of the unidirectional master key MP_{AB} (i.e. the mutual primitive from fax machine A to fax machine B). Knowledge of MP_{AB} allows to decipher all confidential transmissions from A to B.

Attack 2 An algorithm which constructs the unidirectional master key MP_{AX} from A to any machine X from a given master key MP_{AB} using 32 modular multiplications when the identity string ID_A of machine A is known. This algorithm may be viewed as an extension of Attack 1: with only little additional work all confidential transmissions from A to any other fax machine can be deciphered.

Attack 3 An algorithm which constructs the unidirectional master key MP_{XB} from any machine X to the machine B from a given master key MP_{AB} using 32 modular multiplications when the identity string ID_B of machine B and the registered crypt string RCS_{XB} are known.

Attack 4 An algorithm which constructs the unidirectional master key MP_{XY} from any machine $X \neq B$ to any machine Y from a given master key MP_{AB} using 32 modular multiplications when the identity string ID_B of machine B and the registered crypt string RCS_{XY} are known. This algorithm is the extension of Attack 2: if one machine is compromised then all confidential transmissions in the system can be deciphered.

Attack 5 An algorithm which constructs the unidirectional master key MP_{AX} from A to any machine X from two given master keys MP_{AB} and MP_{AC} using 64×10^8 modular multiplications. This algorithm is a 'hard working' version of Attack 2: with more additional work all confidential transmissions from A to any other fax machine can be deciphered even without the knowledge of ID_A.

Attack 6 An algorithm which constructs the unidirectional master key MP_{XB} from any machine X to machine B from two given master keys MP_{AB} and MP_{CB} using 64×10^8 modular multiplications when the registered crypt string RCS_{XB} is known.

Attack 7 A known-plaintext attack which can recover from 6100 consecutive bits (763 consecutive bytes) of plaintext the rest of the fax message (and other fax messages enciphered with the same session key) in 10^7 binary operations using the Massey-Berlekamp algorithm.

Attack 8 A known-plaintext attack which can recover the session key SK from 40 consecutive bits (5 consecutive bytes) of plaintext using 10^9 modular operations and 10^6 bytes of storage. The session key allows to decipher the rest of the fax message (and other fax messages enciphered with the same session key) without additional work.

Attack 9 An algorithm which can recover the unidirectional master key MP_{AB} (i.e. the mutual primitive from fax machine A to fax machine B) from the session key SK, using 10^{10} modulo operations (16-bit) and 10^9 bytes of storage.

Attack 10 An algorithm which can recover the unique cryptographic string $CRYP_A$ of the sending fax machine A from the identity string ID_A and the mutual primitive MP_{AB}, using $8 \cdot 10^9$ modulo operations (16-bit) and $2 \cdot 10^9$ bytes of storage.

Attack 11 An algorithm which can recover the unique cryptographic string $CRYP_B$ of the receiving fax machine B from the identity string ID_B and the mutual primitive MP_{AB}, using $6 \cdot 10^9$ modulo operations (16-bit) and $2 \cdot 10^9$ bytes of storage.

The above attacks are all practical. Suppose one can computes 10^6 modulo operations in one second (which is typical for a 486 PC), then 10^8 operations can be done in 2 minutes and 10^{10} operations within 3 hours. Figure 3 shows the relationship among these attacks, the purpose of each attack, the assumptions for an attack to work and the complexity of each attack.

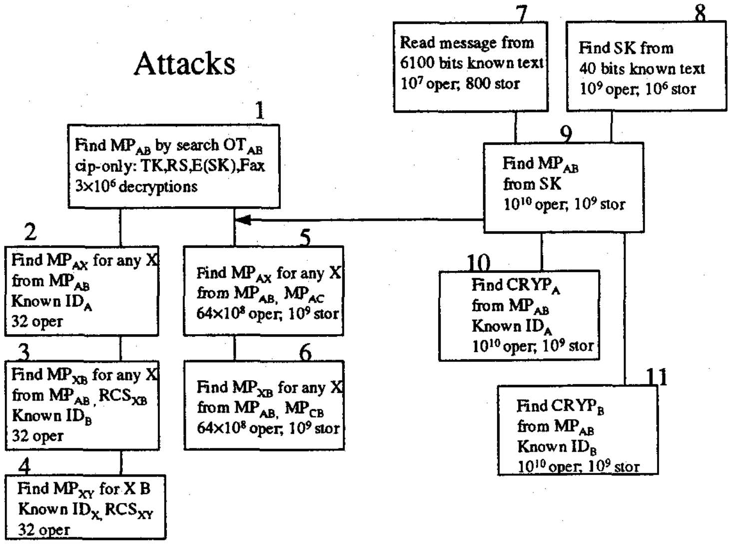

Figure 3. The attacks and their relationship, assumptions and complexities.

4 Attacks on the HKM / HFX cryptosystem

4.1 Attack 1: Exhaustive search on the one time key

This attack is a ciphertext-only attack that breaks the system in 3×10^6 computations of HKM/HFX algorithm by exhaustive search of the one time key OT_{AB}.

1. Intercept the transfer key TK_{AB} at the Registration Mode. Intercept the random string RS, the encrypted session Key ESK and some encrypted fax message at the Automatic Mode.
2. For each possible value (total: 1,000,000) of the 6-digit OT_{AB}, repeat the following:
   - compute (deciphering TK_{AB}) $MP'_{AB} = HKM^{-1}(OT_{AB}, TK_{AB})$
   - compute (deciphering ESK) $SK' = HKM^{-1}(RS, MP'_{AB}, ESK)$
   - compute M'essage = $HFX40^{-1}$(SK', Encrypted fax message), if the result is some intelligible message, i.e., a normally readable fax text, then the trial mutual primitive MP'_{AB} is almost certainly equal to the real MP_{AB} because of the high redundancy of fax message.

Once the MP_{AB} is known, all the future communication from A to B can be recovered. It is an obvious conclusion that the one time key used to protect the MP_{AB} should be at least 12 digits (40 bits) long (so that the system is both exprotable and not obviously weak. But the following analysis will show that even a longer one time key won't provide higher security.

4.2 Attack 2: Finding MP_{AX} from MP_{AB}

We show how to construct the unidirectional master key MP_{AX} from A to any machine X from a given master key MP_{AB} using 32 modular multiplications when the identity string ID_A of machine A is known. From the way of computing Mutual Primitive MP_{AB} shown in Section 2.5, the sending machine fax-number F_A influences only P[0] and P[1], and that the receiving machine number F_B influences only P[2] and P[3]. MP_{AB} can thus be written as

$$MP_{AB} = f_A(F_A) + g_A(F_B) + M_A + CRYP_A \quad (5)$$

where "+" denotes also the digitwise mod 10 addition of two 16-digit strings and where

$$f_A(F_A)[i] = (P[0] \times B[0]^{i+1} \bmod p_0) + (P[1] \times B[1]^{i+1} \bmod p_1) \bmod 10 \quad i = 0,1,\ldots 15 \quad (6)$$

is the sender fax number function at the sending machine A which depends only on F_A and the identity string ID_A,

$$g_A(F_B)[i] = (P[2] \times B[2]^{i+1} \bmod p_2) + (P[3] \times B[3]^{i+1} \bmod p_3) \bmod 10 \quad i = 0,1,\ldots 15 \quad (7)$$

is the receiver fax number function at the sending machine A which depends only on F_B and the identity string ID_A,

$$M_A[i] = \sum_{j=4}^{8} (P[j] \times B[j]^{i+1} \bmod p_j) \bmod 10 \qquad i = 0,1,\ldots 15$$

can be considered as the master secret of the sending machine A which remains invariant for both sending and receiving between machine A and any other machine.

- Suppose MP_{AB} and the identity string ID_A is known. Then functions $f_A(\cdot)$ and $g_A(\cdot)$ are known. Thus, $M_A + CRYP_A$ can be determined from equation (5).
- For any receiving fax machine X the unidirectional master key from A to X,

  $$MP_{AX} = f_A(F_A) + g_A(F_X) + M_A + CRYP_A$$

  can then be obtained in 32 modulo operations.

4.3 Attack 3: Finding MP_{XB} from MP_{AB}

We show how to find the unidirectional master key MP_{XB} from any machine X to the machine B from a given master key MP_{AB} using 32 modular multiplications when the identity string ID_B of machine B and the registered crypt string RCS_{XB} are known.

At the receiving machine B, the registered crypt string RCS_{AB} is computed (see Section 2.7) by

$$RCS_{AB} = MP_{AB} + f_B(F_A) + g_B(F_B) + M_B \quad (8)$$

where $f_B(\cdot)$ and $g_B(\cdot)$ are the sender and receiver number functions of machine B, which depend only on the fax numbers F_A, F_B and the identity string ID_B, and where

$$M_B[i] = \sum_{j=4}^{8} (P[j] * B[j]^{i+1} \bmod p_j) \bmod 10 \qquad i = 0,1,\ldots 15 \quad (9)$$

is the master secret of the receiving machine B which remains invariant for both sending and receiving between machine B and any other machine.

- Suppose MP_{AB} and the identity string ID_B are known. Suppose further that the registered crypt string RCS_{AB} is known (RCS_{AB} is transferred from machine B to machine A at the Register mode and from A to B at the Automatic mode), then M_B can be obtained from equation (8).
- For any sending machine X, suppose the registered crypt string RCS_{XB} is known, then the mutual primitive MP_{XB} can be obtained from the following equation

  $$RCS_{XB} = MP_{XB} + f_B(F_X) + g_B(F_B) + M_B.$$

4.4 Attack 4: Finding MP_{XY}

Once the MP_{XB} for any machine $X \neq B$ is found, then for any machine Y, Attack 2 implies that one can easily find MP_{XY} when identity string ID_X of machine X is known.

Remark

The above attacks showed that the system has a 'virus' effect: suppose one machine, say B, is compromised, then if any machine X starts a transfer with B (i.e., the RCS_{XB} is sent over the channel), then the machine X is also compromised in the sense that any message from X to any other machine Y can be recovered. If an enemy joins the system by owning such a machine B, then he can break into the communication between any other two machines by ciphertext-only attack.

4.5 Attack 5: Finding MP_{AX} from MP_{AB}, MP_{AC} without knowing ID_A

The attacks 2, 3 and 4 are based on the assumption that the unique identity string of a fax machine is known to the attacker. Although one should reasonably consider an 'identity' as accessible information, our attacks can be extended to the case of fax machines with 'secret identity'. The price we shall pay in this case is more computations and the knowledge of a pair of mutual primitives.

Suppose two different MP_{AB} and MP_{AC} are known. For any $X \neq A,C$, MP_{AX} can be obtained without having to know the ID_A. From equation (5), we have

$$MP_{AB} = f_A(F_A) + g_A(F_B) + M_A + CRYP_A$$

$$MP_{AC} = f_A(F_A) + g_A(F_C) + M_A + CRYP_A$$

Their difference, $MP_{AB} - MP_{AC} = g_A(F_B) - g_A(F_C)$ in decimal digits form, can be written as

$$MP_{AB}[i] - MP_{AC}[i] - (P[2] * B[2]^{i+1} \bmod p_2) + (P'[2] * B[2]^{i+1} \bmod p_2)$$
$$= (P[3] * B[3]^{i+1} \bmod p_3) - (P'[3] * B[3]^{i+1} \bmod p_3) \mod 10 \quad i = 0,1,\ldots 15.$$

Unique ID_A string 48 digits (sending machine)													Crypt String 16 digits				
4	4	4	4	4	4	4	2	2	4	4	4	4	4	4	4	2	2
0-3	4-7	8-11	12-15	16-19				28-31	32-35		40-43	44-47					
	+101	+202	+303	+404	+505	+606	+707	+808		+79	+2*79	+3*79	+4*79	+5*79	+6*79	+7*79	+8*79
$+F_A[0]$	$+F_A[1]$	$+F_B[0]$	$+F_B[1]$														
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]

From the formation of P[i],P'[j],B[k], it can be seen that the left side of the above equation depends only on the digits 8-11, 40-43 of ID_A; the right side depends only on the digits 12-15, 44-47 of ID_A. The following method to determine these digits is the well-known 'meet-in-the-middle' attack [6] that is based on the 'birthday paradox'[7].

For each of 10^8 possible values of digits 8-11, 40-43 of ID_A, compute the left side and store them in a sorted table. For each of 10^8 possible values of digits 12-15, 44-47 of ID_A, compute the right side. Note that for the true values, left side equals to the right side. The probability that there are more than two such values is small. This attack needs at most 64×10^8 operations and 10^8 16-digit storage (about 1 Gigabytes). After the digits 8-15, 40-47 of ID_A have been found, one can compute function $g_A(\cdot)$. From now on, every MP_{AX}, $X \neq B,C$, can be easily computed as

$$MP_{AX} = f_A(F_A) + g_A(F_X) + M_A + CRYP_A = g_A(F_X) + MP_{AB} - g_A(F_B).$$

4.6 Attack 6: Finding MP_{XB} from MP_{AB}, MP_{CB} without knowing ID_B

Suppose that MP_{AB}, MP_{CB}, RCS_{AB} and RCS_{CB} are known. Similar to the above attack, we obtain from equation (3) and (8) that

$$RCS_{AB} - RCS_{CB} = MP_{AB} - MP_{CB} + f_B(F_A) - f_B(F_C)$$

which is further reduced to

$$RCS_{AB}[i] - RCS_{CB}[i] - MP_{AB}[i] + MP_{CB}[i] + (P[0] \times B[0]^{i+1} \bmod p_0) - (P'[0] \times B[0]^{i+1} \bmod p_0)$$
$$= (P[1] \times B[1]^{i+1} \bmod p_1) - (P'[1] \times B[1]^{i+1} \bmod p_1) \mod 10 \quad i = 0,1,\ldots 15.$$

Note that the left side of the equation depends only on the digits 0-3, 32-35 of ID_B; the right side depends only on the digits 4-7, 36-39 of ID_B. Use the same meet-in-the-middle attack as above, we find the digits 0-8, 32-39; i.e., we know function $f_B(.)$. From now on, for any machine $X \neq A,C$, if RCS_{XB} is known, then from

$$RCS_{AB} = MP_{AB} + f_B(F_A) + g_B(F_B) + M_B$$
$$RCS_{XB} = MP_{XB} + f_B(F_X) + g_B(F_B) + M_B$$

MP_{XB} can be easily computed as

$$MP_{XB} = MP_{AB} - RCS_{AB} + RCS_{CB} + f_B(F_A) - f_B(F_X).$$

4.7 Attack 7: Recover fax from 6100 bits known plaintext

This is a known-plaintext attack which recovers from 6100 consecutive bits (763 consecutive bytes) of known plaintext the rest of the fax message (and other fax messages enciphered with the same session key).

From the description of Section 2.4.2, the binary fax message sequence $m_0, m_1, \ldots$, is encrypted to the ciphertext sequence $c_0, c_1, \ldots$, as

$$c_i = m_i \oplus x[i \bmod 1021] \oplus y[i \bmod 1019] \oplus z[i \bmod 1013] \quad i=0,1,\ldots$$

Sequences X, Y and Z can be considered as linear feedback shift register (LFSR) sequences [5] with periods 1021, 1019 and 1013 respectively. HFX is therefore an additive stream cipher with keystream being the XOR sum of three LFSR producing X,Y and Z sequences. Let L(X) denote the linear complexity of sequence X. From the theory of LFSR sequences, we know that

$$L(X \oplus Y \oplus Z) \leq L(X) + L(Y) + L(Z) \leq 3050$$

and that the rest of the keystream can be recovered by using the Berlekamp-Massey LFSR synthesis algorithm in less than 3×10^7 binary operations when 6100 bits of the keystream are known, i.e., when 6100 bits of plaintext are known.

4.8 Attack 8: Finding Session Key from 40 bits known plaintext

In this attack, we show that the session key SK can recovered from 40 consecutive bits (5 consecutive bytes) of known plaintext. Suppose that the first 40 bits of plaintext are known, then we know the bits

$$c_i \oplus m_i = (P[0]\, B[0]^{i+1} \bmod p_0) \oplus (P[1]\, B[1]^{i+1} \bmod p_1) \oplus (P[2]\, B[2]^{i+1} \bmod p_2) \quad i = 0,1,\ldots,39$$

that is,

$$(\underset{01}{P[0]} \times \underset{67}{B[0]}^{i+1} \bmod \underset{012}{p_0}) \oplus (\underset{23}{P[1]} \times \underset{89}{B[1]}^{i+1} \bmod \underset{345}{p_1}) = c_i \oplus m_i \oplus (\underset{45}{P[2]} \times \underset{ab}{B[2]}^{i+1} \bmod \underset{678}{p_2}) \qquad i = 0,1,\ldots,39$$

(the numbers indicate which digits of the session key SK influence the numbers P[i],B[j] and the choice of p_k according to the description given in Section 2.4.2). We shall determine these digits by the following 'meet-in-the-middle' attack:

1. For every of the 19 choices of p_2, compute a table of the right side for every value of digits 45, ab; (10^4 40-bit values, 50KBytes).
2. For each p_2, there are $(1000/19) \approx 53$ possible ways to choose digits 678 at the left side so that the choice is consistant with the way of determine p_2, and there are 18×17 possible choices for p_0 and $p1$. For each (p_0, p_1), there are $(1000/18) \approx 56$ possible values of digits 012. There are 100 possible values for digits 3 and 9.

   For every possible values of digits 01236789, compute the left side value. Total number of possibilities is $53 \times 100 \times 56 \times 18 \times 17 < 10^8$.
3. For the true value of the session key, the left side equals to the right side. The probability that there are more than two values yielding such equality is small. (Because $10^8 \times 10^4 < 2^{40}$ and using the extended birthday argument).

The total complexity is a storage of 19×10^4 40-bit vectors, which is less than 1 Mbytes, and about 20×10^8 modulo operations.

4.8.1 Masquerade

Once the session key is known to the attacker, he can use it to encrypt any FAX message of his choice to the receiver. Because these forgery messages will be decrypted correctly by the receiver, then according to the design, the receiver authenticates sender so that the attacker has successfully masqueraded the sender.

4.9 Attack 9: Finding MP_{AB} from Session key SK

The algorithm below can be used to recover the unidirectional master key MP_{AB} from the session key SK, using 10^{10} modulo operations (16-bit) and 10^9 bytes of storage. Suppose the session key SK is known, then ESK – SK is known. Rewrite equation (4) for i=0,1,...,11 as

$$ESK[i] - SK[i] - \sum_{j=0,1,4,5} (P[j]*B[j]^{i+1} \bmod p_j) \bmod 10 = \sum_{j=2,3,6,7,8} (P[j]*B[j]^{i+1} \bmod p_j) \bmod 10$$

Note that the left side depends only on digit 0-7 of mutual primitive MP_{AB}, and the right side depends only on digit 8-15 of MP_{AB}. We shall determine these digits by 'meet-in-the middle':

1. For each of 10^8 possible values of digit 0-7 of MP_{AB}, compute the left side value;
2. For each of 10^8 possible values of digit 8-15 of MP_{AB}, compute the right side and see if it is equal to one of the left side values;
3. The attack outputs the possible values for MP_{AB} for which the left side is equal to the right side.

The complexity of this attack is at most $9\times12\times10^8$ modulo operations and a storage of 12×10^8 decimal digits (about 2^{29} Bytes, 0.5 Gbytes).

4.10 Attack 10: Finding $CRYP_A$ from MP_{AB}

Suppose the mutual primitive MP_{AB} is known. Suppose further that the unique identity string ID_A of the machine A is also known to the attacker. Note that the sender fax number and receiver fax number are public information. The unique crypt string $CRYP_A$ of the sending machine A can be obtained as follows:

Write encryption equation as

$$MP_{AB}[i] = CRYP_A[i] + \sum_{j=0}^{3} (P[j]B[j]^{i+1} \bmod p_j) + \sum_{j=4}^{5} (P[j]B[j]^{i+1} \bmod p_j) + \sum_{j=6}^{8} (P[j]B[j]^{i+1} \bmod p_j) \bmod 10$$

for i = 0,1,...15. Note that the term $\sum_{j=0}^{3}$ is independent of $CRYP_A$ and that the term $\sum_{j=4}^{5}$ depends only on digits 0–7 of $CRYP_A$ and that the term $\sum_{j=6}^{8}$ depends only on digits 8–15 of $CRYP_A$. We obtain a system:

$$MP_{AB}[0] - CRYP_A[0] - \sum_{j=0}^{3} (P[j]\,B[j]^{i+1} \bmod p_j) - \sum_{j=4}^{5} (P[j]\,B[j]^{i+1} \bmod p_j) = \sum_{j=6}^{8} (P[j]\,B[j]^{i+1} \bmod p_j) \bmod 10$$

...

$$MP_{AB}[7] - CRYP_A[7] - \sum_{j=0}^{3} (P[j]\,B[j]^{i+1} \bmod p_j) - \sum_{j=4}^{5} (P[j]B[j]^{i+1} \bmod p_j) = \sum_{j=6}^{8} (P[j]\,B[j]^{i+1} \bmod p_j) \bmod 10$$

$$MP_{AB}[8] - \sum_{j=0}^{3} (P[j]B[j]^{i+1} \bmod p_j) - \sum_{j=4}^{5} (P[j]B[j]^{i+1} \bmod p_j) = CRYP_A[8] + \sum_{j=6}^{8} (P[j]B[j]^{i+1} \bmod p_j) \bmod 10$$

$$MP_{AB}[15] - \sum_{j=0}^{3} (P[j]B[j]^{i+1} \bmod p_j) - \sum_{j=4}^{5} (P[j]B[j]^{i+1} \bmod p_j) = CRYP_A[15] + \sum_{j=6}^{8} (P[j]B[j]^{i+1} \bmod p_j) \quad \bmod 10$$

Note that the left side depends only on digits 0–7 of $CRYP_A$ and that the right side depends only on digits 8–15 $CRYP_A$.

For each of 10^8 values of the first 8 digits of $CRYP_A$, compute the left side value (a 16-digit string); and for each of 10^8 values of the last 8 digits of $CRYP_A$, compute the right side value. Note that for the true value of $CRYP_A$ the left side equals the right side. The probability that there are more than two randomly chosen values will produce such a equality is small. The complexity of this 'meet-in-the-middle' attack is at most $(32+48)\times10^8$ modulo operations and a storage of 16×10^8 digits $\cong$ 1 GigaBytes.

4.11 Attack 11: Finding $CRYP_B$ from MP_{AB}

Suppose that the MP_{AB} is known. Suppose further that the unique identity string ID_B of the receiving machine B is also known. Rewrite equation (4) as

$$RCS_{AB}[i] - MP_{AB}[i] - \sum_{j=0}^{3} (P[j]\, B[j]^{i+1} \bmod p_j) - \sum_{j=4}^{5} (P[j]\, B[j]^{i+1} \bmod p_j) = \sum_{j=6}^{8} (P[j]\, B[j]^{i+1} \bmod p_j) \bmod 10$$

for i = 0,1,...15. The left side depends only on digits 0–7 of $CRYP_B$ and the right side depends only on digits 8–15 of the $CRYP_B$. By applying again the meet-in-the-middle attack, one can find $CRYP_B$ using at most $(32+48)\times10^8$ modulo operations and a storage of 16×10^8 digits $\cong$ 1 GigaBytes.

5 Conclusion

In this paper we showed eleven attacks on the HKM / HFX cryptosystems, which is proposed for standardization at the ITU Telecommunication Standardization Sector Study Group 8 to provide authenticity and confidentiality of FAX messages at a commercial level of security. The basic techniques used in the attacks are "divide and conquer" and "meet-in-the-middle". The complexities of the attacks vary from a few operations to at most 10^{10} integer operations, indicating that the security provided by the HKM / HFX cryptosystem is too low to meet the requirements for an international standard of the ITU, even with the additional feature of free exportability.

6 References

[1] "Security requirements for group s facsimile", International Telecommunication Union, Telecommunication Standardization Sector, Study Group 8, Contribution 59, Source: United Kingdom, April 1994.

[2] "Proposed security system for group s facsimile", International Telecommunication Union, Telecommunication Standardization Sector, Study Group 8, Contribution 60, Source: United Kingdom, April 1994.

[3] "Details of the HKM algorithm and examples of its use", International Telecommunication Union, Telecommunication Standardization Sector, Study Group 8, Delayed Contribution 152, Source: United Kingdom, June 1994.

[4] "Proposed security system for group s facsimile, The HFX40 algorithm", International Telecommunication Union, Telecommunication Standardization Sector, Study Group 8, Delayed Contribution, June 1994.

[5] A. Rueppel, 'Analysis and Design of Stream Ciphers', New York, NY, Springer-Verlag, 1986.

[6] R. Merkle and M. Hellman, 'On the security of multiple encryption', Communications of ACM, 24(7):465-467, 1981.

[7] K. Nishimura and M. Sibuya, 'Probability to meet in the middle', J. Cryptology, Vol. 2, Nr. 1, 1990, pp. 13-22.