Common-Key Cryptosystem with Mixture of Fake Plaintexts

Masayoshi Hayashi and Hiroaki Higaki[*]
Department of Robotics and Mechatronics, Tokyo Denki University, Japan
[*]Corresponding author

***Abstract*—One of the fundamental methods for eavesdroppers to achieve a plaintext from a cryptogram is the brute force attack where possible candidates of decryption keys are exhaustively applied to the decryption algorithm. Here the only reason why the eavesdroppers believe to find the common-key and to achieve the plaintext is that the output of the decryption algorithm is contextually acceptable. According to this fact, this paper proposes a novel common-key cryptosystem where fake plaintexts which are also contextually acceptable are mixed into a cryptogram with the legal plaintext. If an eavesdropper applies a fake common-key to the decryption algorithm, it outputs the fake plaintexts which the eavesdroppers might believe legal. This paper also proposes concrete encryption/decryption algorithm which can be combined with any conventional common-key cryptosystem.**

Keywords—cryptography; common key cryptosystem; fake plaintexts; brute force attack

I. INTRODUCTION

For support enough security in recent network environments, especially including wireless networks where wireless signals are easily overheard by any other wireless nodes including eavesdroppers, cryptography is widely applied. There are two classes of currently available cryptography; common-key and asymmetry-key cryptosystems. For eavesdroppers, wiretapping of cryptogram and estimation of the decryption key is essential for achieving the plaintext illegally since the encryption/decryption algorithms are usually public in these internetworking era. With help of cheaper high-performance computers, widely available encryption/decryption algorithms face the crisis of the brute force attack. Here, a contextually acceptable output of the decryption algorithm is believed to be the original plaintext. Hence, this paper proposes a novel common-key cryptosystem to solve the problem of the brute force attack by making possible for a decryption algorithm to output not only the legitimate plaintext but also fake plaintext to deceive the eavesdroppers.

II. RELATED WORKS

In cryptography for secure transmissions of valuable information called plaintexts from a source computer to a destination one, a source computer translates each plaintext into a cryptogram, the cryptogram is transmitted through networks and a destination computer extracts the plaintext from the cryptogram. Here, a pair of an encryption and a decryption algorithms for translation between a plaintext and a cryptogram provides enough security to make difficult for eavesdroppers to illegally achieve the plaintext from the wiretapped cryptogram.

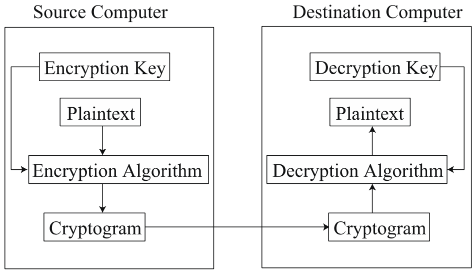

FIGURE I. CRYPTOGRAPHY WITH ENCRYPTION/DECRYPTION KEYS

The encryption and decryption algorithms are usually implemented as software products in currently available various computers connected to open networks such as the Internet. That is, not only the encryption algorithm for translation from a plaintext to a cryptogram but also the decryption algorithm for reverse translation from a cryptogram to a plaintext are public as open software for all possible users including the eavesdroppers. Hence, the provision of enough security currently depends on secret parameters for the algorithms, i.e., most of widely available encryption/decryption algorithms require encryption and decryption keys as their inputs (Figure 1).

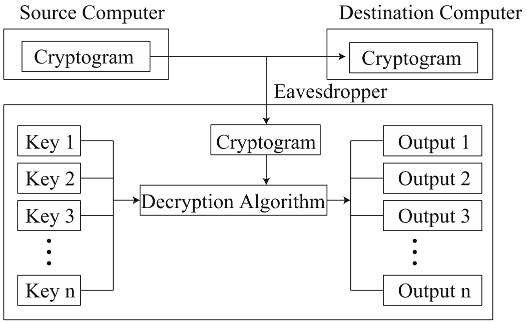

FIGURE II. BRUTE FORCE ATTACK

One of the methods for eavesdroppers to achieve a plaintext from a cryptogram is the brute-force attack. An eavesdropper tries to extract the plaintext by from the cryptogram by applying a decryption algorithm with all possible decryption key candidates (Figure 2). Theoretically, the eavesdropper should try too large number of candidate decryption keys to detect the legal decryption key and achieve the plaintext illegally. Thus, various methods for estimation of the legal decryption key have been developed. By gathering huge numbers of cryptograms transmitted through networks and analyzing them by using cheap but high-performance computers, decryption keys might be estimated depending on some statistical deviation and the currently widely-available cryptosystems might fall into crisis in near future.

Now, consider a case that an eavesdropper tries to achieve the plaintext from a wiretapped cryptogram by the brute force attack. The eavesdropper applies the decryption algorithm to the cryptogram with candidate decryption keys one by one and regards the candidate as the legal decryption key if the output of the decryption algorithm {it seems contextually acceptable}. This criterion is usually too vague; however, it is inevitable since the eavesdropper has only the cryptogram and is impossible to refer the original plaintext. Hence, even if the eavesdropper achieves a contextually acceptable output from the decryption algorithm, it is not always the same as the legitimate plaintext. In addition, if the eavesdropper achieves multiple contextually acceptable output from the decryption algorithm by using different decryption key candidates, it is also impossible to surely select one of them as the legitimate plaintext as shown in Figure 3.

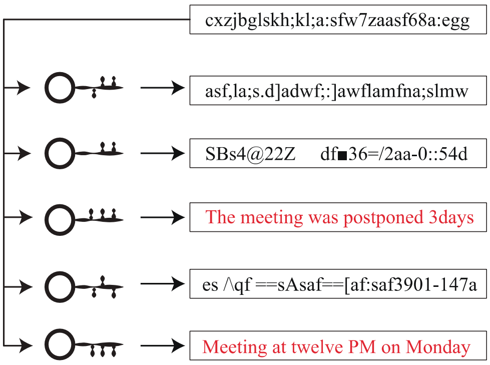
cxzjbglskh;kl;a:sfw7zaasf68a:egg
asf,la;s.d]adwf;:]awflamfna;slmw
SBs4@22Z df■36=/2aa-0::54d
The meeting was postponed 3days
es /\qf ==sAsaf==[af:saf3901-147a
Meeting at twelve PM on Monday

FIGURE III. MULTIPLE CONTEXTUALLY ACCPTABLE OUTPUTS OF DECRYPTION ALGORITHM.

In [2], the one time pad whose decryption algorithm can generate multiple contextually acceptable outputs from a cryptogram has proposed. In its encryption algorithm, a cryptogram *ET* is generated by bit-by-bit exclusive-OR calculation between a plaintext *PT* and a same size common-key K. In its decryption algorithm, the plaintext *PT* is achieved by bit-by-bit exclusive-OR calculation between the cryptogram *ET* and K. Here, number of possible candidate's decryption keys is $2^{|K|}$ and all possible |*PT*| bit-length outputs is generated by applying the decryption algorithm with all the possible decryption candidates. Hence, numerous numbers of contextually acceptable outputs are surely expected to be generated. For example, all the possible |*PT*| bit-length text files are generated by the decryption algorithm and it is impossible for the eavesdroppers to determine which is the legitimate plaintext. However, since the outputs of the encryption algorithm take over the statistical deviation of the plaintexts, it may be possible to estimate the encryption key by analyzing numbers of outputs. In cryptography for secure transmissions of valuable information called plaintexts from a source computer to a destination one, a source computer translates each plaintext into a cryptogram, the cryptogram is transmitted through networks and a destination computer extracts the plaintext from the cryptogram. Here, a pair of an encryption and a decryption algorithms for translation between a plaintext and a cryptogram provides enough security to make difficult for eavesdroppers to illegally achieve the plaintext from the wiretapped cryptogram. The encryption and decryption algorithms are usually implemented as software products in currently available various computers connected to open networks such as the Internet. That is, not only the encryption algorithm for translation from a plaintext to a cryptogram but also the decryption algorithm for reverse translation from a cryptogram to a plaintext are public as open software for all possible users including the eavesdroppers. Hence, the provision of enough security currently depends on secret parameters for the translation from a cryptogram to a plaintext are public as open software for all possible users including the eavesdroppers. Hence, the provision of enough security currently depends on secret parameters for the algorithms, i.e., most of widely available encryption/decryption algorithms require encryption and decryption keys as their inputs (Figure 1).

One of the methods for eavesdroppers to achieve a plaintext from a cryptogram is the brute-force attack. An eavesdropper tries to extract the plaintext by from the cryptogram by applying a decryption algorithm with all possible decryption key candidates (Figure 2). Theoretically, the eavesdropper should try too large number of candidate decryption keys to detect the legal decryption key and achieve the plaintext illegally. Thus, various methods for estimation of the legal decryption key have been developed. By gathering huge numbers of cryptograms transmitted through networks and analyzing them by using cheap but high-performance computers, decryption keys might be estimated depending on some statistical deviation and the currently widely-available cryptosystems might fall into crisis in near future.

III. PROPOSAL

This paper proposes a pair of algorithms for a common-key cryptosystem with mixture of a legitimate plaintext and a fake plaintext in order for more secure communication. In a source computer, an encryption algorithm translates a legitimate and a fake plaintext with a common-key shared with a destination computer into a cryptogram. On the other hand, in the destination computer, a decryption algorithm extracts the legitimate plaintext from the cryptogram by using the same common-key. The decryption algorithm also extracts the fake plaintext from the cryptogram by using a certain available key (Figure 4). Hereafter, we call it a fake common-key.

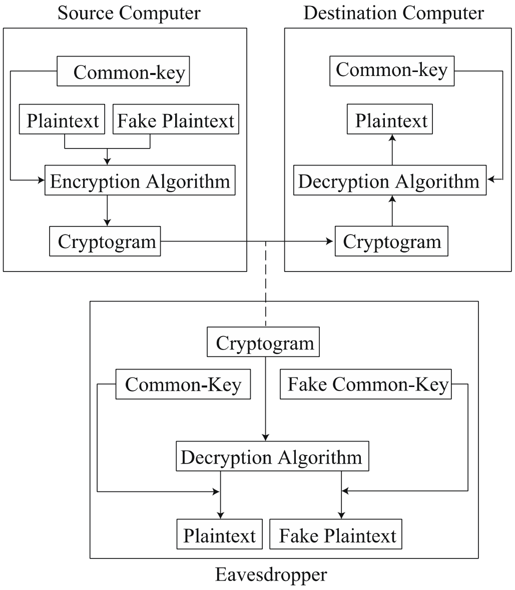

FIGURE IV. OUR PROPOSAL COMMON-KEY CRYPTOSYSTEM WITH MIXTURE OF FAKE PLAINTEXTS.

Same as most of widely available common-key and asymmetric-key cryptosystems, encryption and decryption algorithms are assumed to be open. Hence, an eavesdropper who tries to apply the decryption algorithm to the wiretapped cryptogram with the common-key by accident gets the legitimate plaintext. However, the eavesdropper gets the fake plaintext by applying the decryption algorithm to the cryptogram with the fake common-key. The eavesdropper may believe the achieved fake plaintext to be legitimate and terminate the trials decrypting the cryptogram without achieving the legitimate plaintext. Even though the eavesdropper continues the trials and achieves both the fake and the legitimate plaintexts, it is impossible for the eavesdropper to distinguish them. As a result, our proposal makes difficult for the eavesdropper to achieve the legitimate plaintext (Figure 5).

[Cryptosystem with Mixture of Fake Plaintexts]

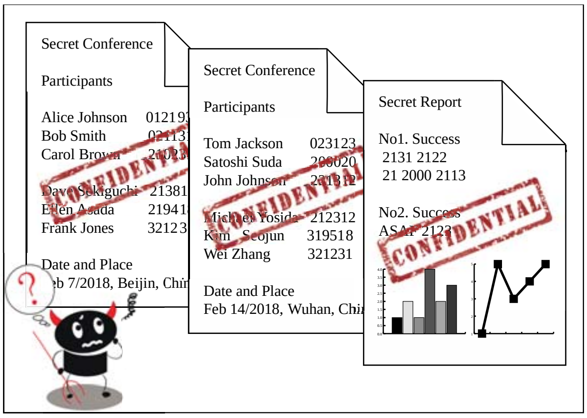

FIGURE V. EFFECTS OF MIXTURE OF FAKE PLAINTEXTS

Let *PT*, *FPT*, K_e and K_d be a legitimate plaintext, a fake plaintext, an encryption key and a decryption key. The following pair of an encryption E and a decryption D algorithms are called algorithms for a cryptosystem with mixture of fake plaintext. Here, FK_e and FK_d are fake encryption and decryption keys, respectively.

$D(ET, K_d) = PT$ and $D(ET, K_d) = FPT$ where $ET := E(PT, FPT, K_e)$

Same as other widely available conventional common-key cryptosystems, it is assumed that a common-key $K := K_e = K_d$ is safely delivered in advance to both the source and the destination computers. Or, same as other widely available conventional asymmetric-key cryptosystems, it is assumed that a decryption-key K_d is strictly kept secret by the destination computer while an encryption-key K_e is publicly delivered possibly through networks. On the other hand, the fake encryption-key FK_e is implicitly generated in the encryption algorithm E. That is, FK_e is generated and used for encrypting the fake plaintext *FPT* in E; however, an explicit output of E is only an encrypted-text (cryptogram) *ET*. FK_e is never used out of E and is never transmitted through any network. In addition, only the existence of FK_d is important for deceiving eavesdroppers by extraction of the fake plaintext *FPT* from *ET* in the decryption algorithm D. Since FK_d is expected to be applied by the eavesdroppers by accident, is never transmitted through any network either. Therefore, the fake common-key $FK := FK_e = FK_d$ in common-key cryptosystems and the fake encryption FK_e and decryption FK_d keys in asymmetric-key cryptosystems never become security flaws. As a concrete encryption and decryption algorithms for common-key cryptosystems with mixture of a fake plaintext, this paper proposes a method concatenating {it sub-cryptograms} which are outputs of a conventional encryption algorithm E with inputs *PT* and *FPT*. The concatenation order is determined only by the common-key K in order to conceal the concatenation order from eavesdroppers. At this time, since a fake common-key *FK* generated in the encryption algorithm E never contradicts the concatenation order, the fake plaintext *FPT* is surely extracted in the decryption algorithm D by using *FK*. The proposed encryption and decryption algorithms are as follows where E' and D' are an encryption and a decryption algorithms of any conventional common-key cryptosystem, respectively.

[Encryption Algorithm E] (Figures 6 and 7)

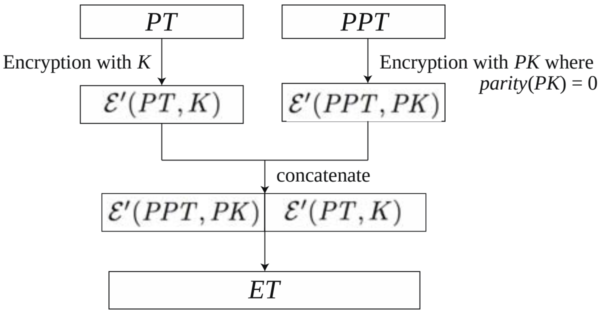

FIGURE VI. ENCRYPTION ALGORITHM (IN CASE OF parity(K) = 0).

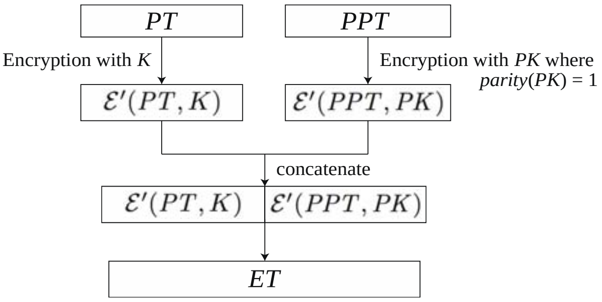
PT
PPT
Encryption with K
Encryption with PK where parity(PK) = 1
$\mathcal{E}'(PT, K)$
$\mathcal{E}'(PPT, PK)$
concatenate
$\mathcal{E}'(PT, K)$ $\mathcal{E}'(PPT, PK)$
ET

FIGURE VII. ENCRYPTION ALGORITHM (IN CASE OF parity(K) = 1)

1) A source computer C_s calculates a binary-parity parity(K) of a common-key K.

2) C_s translates a legitimate plaintext PT to a sub-cryptogram E'(PT, K) by applying E' with K.

3) C_s generates a fake common-key FK satisfying parity(FK)=parity(K)

4) C_s translates a fake plaintext FPT to another sub-cryptogram E'(FPT, FK) by applying E' with FK.

5) C_s generates a encrypted-text (cryptogram) ET by concatenation of E (PT, K) and E'(FPT, FK). The concatenation order is determined by parity(K) as follows where + is a concatenation operator:

a) ET∶= E'(PT, K) + E'(FPT, FK) if parity(K) = 0.

b) ET∶= E'(FPT, FK) + E'(PT, K) if parity(K) = 1.

[Decryption Algorithm D] (Figure 8)

1) A destination computer C_d calculates a binary-parity parity (K) of a common-key K.

2) C_d divides ET into the same size ET [0] and ET [1].

3) C_d extracts the legitimate plaintext D' (ET [parity(K)], K) from ET [parity (K)] by applying D' with K.

[Property]

If the decryption algorithm D is applied to the encrypted-text *ET* with the fake common-key *FK*, the fake plaintext *FPT* is extracted instead of *PT*.

That is, D' (*ET* [parity(*FK*)], *FK*) = *FPT* is satisfied.

Hence, an eavesdropper under a brute-force attack extracts the fake plaintext by accidently using the fake common-key as discussed in this section.

IV. CONCLUDING REMARKS

This paper proposes a novel common-key cryptosystem with mixture of fake plaintexts. In order to support enough security even against the brute force attack, the proposed decryption algorithm generates not only the legitimate plaintext but also fake plaintexts which are contextually acceptable and the eavesdroppers cannot determine which output is legal. The proposed algorithm can combine with any conventional common-key cryptosystem providing enough statistical difficulty and computational complexity for common-key estimation.

In future work, we evaluate the proposed method by possibility of illigal achievement of legitimate plaintext in comparison with the conventional common-key cryptosystems.

REFERENCES

[1] Daemen, J., RijnMen, V., “The Rijndael Block Cipher, ” AES proposal, First AES Candidate Conference (AES1), (1998).

[2] Shannon, C.E., “Communication Theory of Secrecy Systems, ” IEEE Press, p. 84--143 (1949).