ISSN1000 9825 Journal of Software 软件学报 2000,11(5):584~589

# Abuses of Ajtai-Dwork Cryptosystem*

ZHAO Zhu

(Department of Computer Science Zhangye Teachers' College Zhangye 734000)
E-mail: zhaozhu@public.lz.gs.cn

**Abstract** Ajtai and Dwork have introduced a probabilistic public-key encryption scheme which is secure under the assumption that a certain computational problem on lattices is hard on the worst-case. In this paper, the author demonstrates how Ajtai-Dwork cryptosystem can be abused. Using this kind of abuses, users can communicate secrets in a key escrowed Ajtai-Dwork cryptosystem without fearing that their secrets will be revealed later by reconstructing their escrowed private-keys. However, it is also shown that users have to trust their implementers because unscrupulous implementers of Ajtai-Dwork cryptosystem may leak their private-keys without their awareness. The author shows how one can make Ajtai-Dwork cryptosystem abuse-free.

**Key words** Subliminal channel, abuse of cryptosystem, key escrow.

Simmons[1] discovered in 1983 that in several authentication schemes (e.g., DSS and Schnorr's identification scheme) one could hide covert data in the authenticator, which he called a *subliminal channel*. His example is related to two prisoners who are communicating authentical messages in full view of a warden, who is able to read the messages. The subliminal channel consists in hiding a message through the authentication scheme such that the warden cannot detect its use nor read the hidden part. In this paper, we will show that it is very easy to build a subliminal channel in the Ajtai-Dwork cryptosystem.

Micali introduced *key escrow cryptosystems* in Ref. [2]. In these cryptosystems, the private-key is broken up into pieces and distributed to different authorities. The authorities can get together and reconstruct the private-key. Key escrow guarantees that the police can eavesdrop on all conversations or personal data files even though they are encrypted. Of course, a cryptography user gains nothing from key escrow systems at all. He has to trust the escrow agents' security procedures, as well as the integrity of the people involved. However, we will show that even if a user surrenders his private key for Ajtai-Dwork cryptosystem to key escrow agents, he still can abuse Ajtai-Dwork cryptosystem and communicate under the government's very nose without having to worry that it would be detected. We will show how one can make Ajtai-Dwork cryptosystem abuse-free.

We close this section by introducing some notations we will use. $\mathcal{N}$, $\mathcal{Z}$, and $\mathcal{R}$ are the set of natural numbers, the set of integers, and the set of real numbers respectively. $\{0,1\}^*$ is the set of (finite) binary strings, $\{0,1\}^n$ is the set of binary strings of length $n$. The length of a string $x$ is denoted by $|x|$. For a string $x\in\{0,1\}^*$ and an integer number $n\geqslant 1$, $x[1..n]$ denotes the initial segment of length $n$ of $x$ ($x[1..n]=x$ if $|x|\leqslant n$) and $x[i]$ denotes the $i$th bit of $x$, i.e., $x[1..n]=x[1]\ldots x[n]$.

Let $\mathbf{b}_1,\ldots,\mathbf{b}_n$ be a basis of $\mathcal{R}^n$. Then a *lattice* in $\mathcal{R}^n$ is a set of the form

$$L=L(\mathbf{b}_1,\ldots,\mathbf{b}_n)=\left\{\sum_{i=1}^{n}x_i\mathbf{b}_i : x_i\in\mathcal{Z}\right\},$$

---

* **ZHAO Zhu** was born in 1968. He is a lecturer of Department of Computer Science, Zhangye Teachers' College. His research interests are application of database and cryptography.

Manuscript received 1997-12-12, accepted 1998-03-30.

and we say that $\mathbf{b}_1,\ldots,\mathbf{b}_n$ is a basis of $L$. For a vector $\mathbf{x}=(x_1,\ldots,x_n)\in\mathscr{R}^n$, the Euclidean norm and the maximum norm $L_\infty$ of $\mathbf{x}$ are $\|\mathbf{x}\|=\sqrt{\sum_{i=1}^{n}x_i^2}$ and $\|\mathbf{x}\|_\infty=\max_{1\le i\le n}|x_i|$ respectively.

## 1 Ajtai-Dwork Cryptosystem

Ajtai and Dwork[3] have introduced a public-key encryption scheme which is secure unless the worst case of the following lattice problem can be solved in polynomial time: find the shortest nonzero vector in an $n$-dimensional lattice $L$ where the shortest vector $\mathbf{u}$ is unique in the sense that any other vector whose length is at most $n^c\|\mathbf{u}\|$ is parallel to $\mathbf{u}$. Roughly speaking, an instance of their cryptosystem is a collection of hidden hyperplanes $H_i=\{\mathbf{v}:\langle\mathbf{u},\mathbf{v}\rangle,\mathbf{v}\in\mathscr{R}^n\}$ for $i\in\mathscr{Z}$, where $\mathbf{u}$ is the unique shortest vector in a lattice $L$ and $\langle\mathbf{u},\mathbf{v}\rangle$ denotes the inner product of $\mathbf{u}$ and $\mathbf{v}$. The private-key is the unique shortest vector $\mathbf{u}$, and the public-key is a method of generating a point guaranteed to be near one of the hyperplanes in the collection. The public-key is chosen so as not to reveal the collection of hyperplanes. Encryption is bit by bit: zero is encrypted by using the public-key to find a random vector $\mathbf{v}\in\mathscr{R}^n$ near one of the hyperplanes—— the ciphertext is $\mathbf{v}$; one is encrypted by choosing a random vector $\mathbf{v}$ uniformly in $\mathscr{R}^n$——the ciphertext is simply $\mathbf{v}$. Decryption of a ciphertext $\mathbf{x}$ is performed using the private key to determine the distance of $\mathbf{x}$ to the nearest hidden hyperplane. If this distance is sufficiently small, then $\mathbf{x}$ is decrypted as zero; otherwise $\mathbf{x}$ is decrypted as one. In their encryption scheme, there is a negletable probability that we decrypt the ciphertext of 1 as 0. Goldreich, Goldwasser, and Halevi[4] modified the encryption method of Ajtai and Dwork and made Ajtai-Dwork cryptosystem error-free. In what follows, we will formally describe the error-free Ajtai-Dwork cryptosystem of Ref. [4]. As in Ref. [4], we present the scheme in terms of real numbers with infinite precision. In reality, one uses approximations (i.e., to $n$-bit binary expansions). The protocol is as follows.

**1. Common parameters.** Given a security parameter $n$, we let $m=n^3$, $R=2^{O(n\log n)}$, and $r=n^{-3}$. We denote by $\mathbf{B}$ (big) the $n$-dimensional sphere of radius $R$, and by $\mathbf{S}$ (small) the $n$-dimensional sphere of radius $r$.

**2. Private-key.** The private-key is a uniformly chosen vector in the $n$-dimensional unit sphere. We denote this vector by $\mathbf{u}$.

**3. Public-key.**

- Select $\mathbf{a}_1,\ldots,\mathbf{a}_m$ uniformly from the set of vectors $\{\mathbf{x}\in\mathbf{B}:\langle\mathbf{x},\mathbf{u}\rangle\in\mathscr{Z}\}$.
- For $i=1,\ldots,m$, select $\delta_{i,1},\ldots,\delta_{i,n}$ uniformly in $\mathbf{S}$, and set $\delta_i=\sum_j\delta_{i,j}$.
- Set $\mathbf{v}_i=\mathbf{a}_i+\delta_i$ for $i=1,\ldots,m$.
- Let $i_0$ be the smallest $i$ for which the width of the parallelepiped spanned by $\mathbf{v}_{i+1},\ldots,\mathbf{v}_{i+n}$ is at least $n^{-2}\cdot R$. (By Ref. [3], with overwhelmingly high probability $i_0$ exists and is smaller that $m/2$.) For $j=1,\ldots,n$, let $\mathbf{w}_j=\mathbf{v}_{i_0+j}$, and denote by $P(\mathbf{w}_1,\ldots,\mathbf{w}_n)$ the parallelepiped spanned by $\mathbf{w}_1,\ldots,\mathbf{w}_n$.
- Pick $i_1$ uniformly at random from all the indices $i$ for which $\langle\mathbf{a}_i,\mathbf{u}\rangle\in 2\mathscr{Z}+1$. That is, $i_1$ is selected so that $\langle\mathbf{a}_{i_1},\mathbf{u}\rangle$ is an odd integer. We note that such an index exists with probability around $1-2^{-m}$.

The public-key consists of the sequence of vectors $\mathbf{v}_1,\ldots,\mathbf{v}_m$ and the integers $i_0,i_1\in\{1,\ldots,m\}$.

**4. Encryption.** The encryption is bit by bit. To encrypt a 0, we uniformly select $b_1,\ldots,b_m\in\{0,1\}^m$, and reduce the vector $\mathbf{v}'=\sum_{i=1}^{m}b_i\cdot\mathbf{v}_i$ modulo the parallelepiped $P(\mathbf{w}_1,\ldots,\mathbf{w}_n)$. That is, find a vector $\mathbf{v}\in P(\mathbf{w}_1,\ldots,\mathbf{w}_n)$ such that $\mathbf{v}'=\mathbf{v}+\sum_{i=1}^{n}c_i\cdot\mathbf{w}_i$ where $c_i$ are all integers. The ciphertext is the vector $\mathbf{v}$. To encrypt a 1 we uniformly select $b_1,\ldots,b_m\in\{0,1\}^m$, and reduce the vector $\frac{1}{2}\mathbf{v}_{i_1}+\sum_{i=1}^{m}b_i\cdot\mathbf{v}_i$ modulo the parallelepiped $P(\mathbf{w}_1,\ldots,$

$\mathbf{w}_n$). And the ciphertext is the reduced vector $\mathbf{v}$.

**5. Decryption.** Given a ciphertext $\mathbf{v}$ and the private-key $\mathbf{u}$, we compute $r=(\mathbf{v},\mathbf{u})$. Decrypt the ciphertext as a 0 if $r$ is within 1/4 of some integer and decrypt it as a 1 otherwise.

Note that the above Ajtai Dwork cryptosystem is a theoretically important system though not a practical one (the construction of the system seems to be inefficient). It is the first cryptosystem whose security is based on the worst-case intractability of certain lattice problems. In the past, all other cryptosystems like RSA and DLP are based on the average case intractability of certain problems like factorization and discrete logarithm computation.

## 2 Abuses of Ajtai-Dwork Cryptosystem

Let us first describe the following situation: Alice and Bob work for a criminal or terrorist organization, and they want to exchange some secrets using Ajtai-Dwork cryptosystem. But according to some laws, they must surrender their private-keys to some key escrow agents. So at some later time, if their keys are revealed, all their communications will be decrypted and they will be put into prison. If the authentication or signature system allows for a subliminal channel, then they can communicate some innocuous messages and put their secrets in the subliminal channel. However, for Ajtai-Dwork cryptosystem, they do not need the subliminal channel at all. They can abuse the system and communicate secrets without being detected. The subliminal channels usually have a small capacity (see, e.g., Ref. [1]). We will show that one can communicate as many secrets as he wants by abusing Ajtai-Dwork cryptosystem without being detected.

In order to abuse Ajtai Dwork cryptosystem, Alice and Bob will first agree on an "abuse-key" $k_a\in\{0,1\}^*$, which should be long enough and random. Now we demonstrate how Alice sends a $|k_a|$-length secret to Bob through Ajtai-Dwork cryptosystem without being detected. In general the protocol looks like this.

1. Assume that Alice wants to send a $|k_a|$-length secret $s$ to Bob.
2. Alice XORs the plaintext $s$ with the abuse-key $k_a$, that is, Alice puts $c=s\otimes k_a$.
3. Alice generates a $|k_a|$-length innocuous message $x$.
4. For each $i\leqslant|k_a|$, Alice uses Bob's public key to check random encryptions of $x[i]$ until an $\mathbf{e}_i$ is found such that

$$\#(\mathbf{e}_i)=\begin{cases}\text{even} & \text{if } c[i]=0\\ \text{odd} & \text{if } c[i]=1\end{cases},$$

where $\#(\mathbf{e}_i)$ denotes the number of 1s in all components of $\mathbf{e}_i$. Note that the odds of an encryption of $x[i]$ having the above property are 1 in 2. Hence Alice can find $\mathbf{e}_i$ by trying 2 encryptions of $x[i]$ on the average.

5. After Bob gets the ciphertext, he reconstructs the secret $s$ simply by counting the numbers of 1s in the ciphertext of $x[i]$'s for all $i\leqslant|k_a|$ and XORing them with $k_a[i]$'s.

It is straightforward to see that the above abuses of Ajtai-Dwork cryptosystem cannot be detected by a warden. Moreover, even if the warden knows the private-key of Bob, he is still unaware that the cryptosystem has been abused!

Our above example shows that a user of Ajtai-Dwork cryptosystem does not need to worry about the key escrow policy. Alice can still have her own privacy even though Bob's private-key has been escrowed and her communications might be eavesdropped by distrustful key escrow agents.

Now we begin to show how a manufacturer (we will call him Mallory) of Ajtai-Dwork cryptosystem implementers uses above abuses of Ajtai-Dwork cryptosystem to leak his customer's private-keys.

1. Mallory puts his implementation of Ajtai-Dwork cryptosystem in a tamperproof VLSI chip, so that no one can examine its inner workings. He embeds our above abuses of Ajtai Dwork cryptosystem in his

implementation. That is, he chooses a long enough random abuse-key $k_a$ (which should be at least as long as commonly used public cryptographic keys).

2. Mallory distributes the chips to his customers, e.g., Alice, Bob, and everyone else who wants them.

3. Every time when Alice sends a message to Bob, the chip reads Alice's private-key $k_A$, and XORs it with the abuse-key $k_a$, that is, let $c=k_A\otimes k_a$. Then the chip encrypts the message and embeds $c$ in the encryption according to our above abuse protocol of Ajtai-Dwork cryptosystem.

4. Alice sends the encrypted message to Bob.

5. Mallory eavesdrops the communications between Alice and Bob. He computes the $c$ from the encrypted message and, because he knows the abuse-key $k_a$, decrypts Alice's private-key $k_A$.

The above attack against Ajtai-Dwork cryptosystem can be easily overcome because Alice can save her private-key in a safe place and does not let the chip read it when she encrypts a message for others. Of course, the chip may be so designed that it will remember Alice's private-key when it decrypts some encrypted messages received by Alice and leak Alice's private-key in the next time when it encrypts a message for Bob. But this attack can also be overcome by using different chips for encryption and decryption. However, every time when Alice wants to sign a message, the chip must read her private signature key, whence the chip can leak her private signature key. It is even worse that even if Alice knows what is happening, she cannot prove it because she does not know the abuse-key $k_a$.

In this section, we will show how to make Ajtai-Dwork cryptosystem abuse-free. Roughly speaking, in order to prevent Alice from abusing the cryptosystem, every time when Alice wants to send a message to Bob, she must send the encrypted message to the censoring warden first. Then the censoring warden perturbs the encrypted message a little and sends the resulting encryption to Bob. More precisely, the protocol looks as follows.

1. Alice generates an encryption $\mathbf{v}'$ of $b\in\{0,1\}$ and sends $\mathbf{v}$ to the censoring warden.

2. The censoring warden chooses uniformly at random a small vector $\delta$ from $\mathbf{S}$, and let $\mathbf{v}=\mathbf{v}'+\delta$. The warden then sends $\mathbf{v}$ to Bob. Note that $\mathbf{S}$ is the $n$-dimensional sphere of radius $n^3$ and $n$ is the security parameter.

3. Bob decrypts the ciphertext $\mathbf{v}$ according to the standard Ajtai-Dwork protocol.

It is easy to check that the above modified Ajtai-Dwork cryptosystem is abuse-free if the warden does not abuse the system. Using the above method, key escrow agents can prevent a user from sending any secret message which they cannot recover after reconstructing the private-key of the user by perturbing the encryption a little every time (which might be very expensive!). If Alice does not trusts her implementers, she can use a software which she trusts to perturb the encryption a little every time before sending the encryption to Bob. Then she can prevent the implementers from leaking her private-key.

## 3 Abuses of Other Probabilistic Cryptosystems

In this section, we show that our attack against Ajtai-Dwork cryptosystem in the last section can also be used to attack other probabilistic cryptosystems like Goldreich-Goldwasser-Halevi cryptosystem[6], McEliece cryptosystem[6], etc.

Recently, Goldreich, Goldwasser, and Halevi[5] have introduced a cryptosystem which is based on the lattice reduction problems. Later in this paper, we will call this system GGH cryptosystem. The security of their system is based on the following conjecture: it is difficult to find closest vectors in a lattice to a given point (CVP).

Roughly speaking, the idea underlying the construction of GGH cryptosystem is based on the following one way trapdoor function. Given any basis for a lattice, it is easy to generate a vector which is close to a lattice

point (i.e., by taking a lattice point and adding to it a small error point). However it seems hard to return from this "close-to-lattice" vector to the original lattice point (given an arbitrary lattice basis). Thus, the operation of adding a small error vector to a lattice point can be thought of as a one-way computation. It seems that different bases of the same lattice yield a difference in the ability to find close lattice points to arbitrary vectors in $\mathscr{R}^n$. Therefore we can use this fact to introduce a trapdoor in the above one-way function. That is, the trapdoor information is a basis of a lattice which allows very good approximation of the closest lattice point problem. Thus, we use two different bases of the same lattice. One basis is the public-key which is used to encrypt messages, and the other basis is the private-key which is used to decrypt messages. Generally, the protocol is as follows.

**1. Common parameter.** Choose a security parameter $n\in\mathscr{N}$ and an error probability $\varepsilon$.

**2. Private-key.** The private-key is a uniformly chosen (at random) $n\times n$ integral matrix $S=(\mathbf{s}_1^T,\ldots,\mathbf{s}_n^T)$ such that $L(\mathbf{s}_1,\ldots,\mathbf{s}_n)$ is a full-rank lattice in $\mathscr{Z}^n$.

**3. Public-key.** Let $\sigma$ be a positive real number which is less than $\{\gamma\sqrt{8\ln(2n/\varepsilon)}\}^{-1}$, where $\gamma/\sqrt{n}$ is the maximum norm $L_\infty$ of the rows in $S^{-1}$. Then the public-key is a pair $(P,\sigma)$ where $P=(\mathbf{p}_1^T,\ldots,\mathbf{p}_n^T)$ is an $n\times n$ integral matrix which is obtained by picking a "random unimodular transformation" of the private-key $S$. The "random unimodular transformation" is obtained by multiplying many "elementary matrices" to $S$. After the transformation, we should usually have the following property.

$\prod_{i=1}^{n}\|\mathbf{s}_i^*\|$ is small and $\prod_{i=1}^{n}\|\mathbf{p}_i^*\|$ is large, where $\mathbf{s}_i^*$ and $\mathbf{p}_i^*$ are the $i$th rows in $S^{-1}$ and $P^{-1}$ respectively.

**4. Encryption.** On input message $s_1\ldots s_m$ and the public-key $(P,\sigma)$, we first apply some (randomized) encoding function $\mathbf{v}_i\leftarrow Enc(s_i)$ to encode $s_i$ as a vector $\mathbf{v}_i\in\mathscr{Z}^n$. (Note that we let $Enc$ and $Dec$ denote a pair of public and easy to compute functions such that $Dec(Enc(s))=s$ for all $s$.) Now pick uniformly at random an error vector $\mathbf{e}\in\mathscr{R}^n$ such that each entry in it has zero-mean and variance $\sigma^2$. Let $\mathbf{c}_i=f_P(\mathbf{v}_i,\mathbf{e})=P\mathbf{v}_i+\mathbf{e}$ be the ciphertext of $s_i$.

**5. Decryption.** To decrypt $\mathbf{c}_i$, we first let $\mathbf{v}_i\leftarrow P^{-1}S\lceil S^{-1}\mathbf{c}_i\rfloor$, where $\lceil\cdot\rfloor$ is the "round off" operation. We then extract the message $s_i$ by letting $s_i=Dec(\mathbf{v}_i)$.

After the description of GGH cryptosystem, it is easy to see that our attack against Ajtai-Dwork cryptosystem in the last section can be used to attack GGH also. That is, in order to send a $|k_a|$-length (where $k_a$ is the abuse-key as in the last section) covert message $x$, first generate an innocuous $|k_a|$-block message $s_1\ldots s_{|k_a|}$. Then choose the error vectors $\mathbf{e}_i$ in such a way that the encryption $\mathbf{c}_i$ of $s_i$ satisfies the following condition.

$$\#(\mathbf{c}_i)=\begin{cases}\text{even} & \text{if } k_a[i]\otimes x[i]=0\\ \text{odd} & \text{if } k_a[i]\otimes x[i]=1\end{cases}.$$

After receiving the ciphertext $\mathbf{c}_1\ldots\mathbf{c}_{k_a}$, it is easy to recover the covert message $x$. The same technique as in the last section can also be used by unscrupulous implementers of GGH cryptosystem to leak user's private-keys. However, our method of making Ajtai-Dwork cryptosystem abuse-free can also be used to make GGH cryptosystem abuse-free. That is, every time when Alice wants to send a message to Bob, she must send the encrypted message to the censoring warden first. The censoring warden perturbs the encrypted message a little by adding another random vector $\mathbf{e}'\subset\mathscr{R}^n$ to the ciphertext $\mathbf{v}$ such that each entry in $\mathbf{e}'$ has zero-mean and variance $\sigma'^2$. And then the warden sends the resulting encryption to Bob. Of course, after adding the random vector $\mathbf{e}'$, the error probability $\varepsilon$ of the system will change a little.

## 4 Remarks

The reader may think that our attack against Ajtai-Dwork cryptosystem is similar to the *shadow public-key cryptosystem* presented by Kilian and Leighton[7]. An example of their shadow public-key cryptosystem is as

follows. In order to cheat key escrow agents, a user called Alice of RSA cryptosystem maintains two key pairs $(s_1, p_1)$ and $(s_2, p_2)$ where $s_i$ and $p_i$ are private keys and public-keys respectively. But she only surrenders her private key $s_2$ to key escrow agents. When a user called Bob wants to send Alice a covert message $x$, he first encrypts the message with Alice's public-key $p_1$ and gets $x_1=E_{p_1}(x)$, then he encrypts $x_1$ again with Alice's public-key $p_2$ and gets $x_2=E_{p_2}(x_1)$. Bob sends Alice the twice-encrypted message $x_2=E_{p_2}(E_{p_1}(x))$. After getting $x_2$, Alice decrypts it twice with her private-keys $s_1$ and $s_2$ and gets the covert message $x=D_{s_1}(D_{s_2}(E_{p_2}(E_{p_1}(x))))$. However, when key escrow agents reconstruct Alice's private-key $s_2$, they can only get the message $x_1$. The problem of this kind of attack is that $x_1$ usually is not a readable message (random), whence key escrow agents will know that Bob must have sent some covert message to Alice. But in our attack against Ajtai-Dwork cryptosystem, we do not have this kind of problem. We hide the covert message in the randomness of the ciphertext instead of encrypting the message twice. And after reconstructing Alice's private-key, key escrow agents will get a readable message and cannot prove that Bob has sent a covert message to Alice.

## References


1 Simmons G J. Verification of treaty compliance revisited. In: Proceedings of the 1983 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 1983. 61~66

2 Micali S. Fair public-key cryptosystem. In: Advances in Cryptology, Proceedings of Crypto'92. Lecture Notes in Computer Science 740, Springer Verlag, 1992. 113~138

3 Ajtai M, Dwork C. A public key cryptosystem with worst case/average-case equivalence. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing. El Paso, 1997. 284~293

4 Goldreich O, Goldwasser S, Halevi S. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In: Advances in Cryptology, Proceedings of Crypto'97. Lecture Notes in Computer Science, Springer Verlag, 1997

5 Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problems. In: Advances in Cryptology, Proceedings of Crypto'97. Lecture Notes in Computer Science, Springer-Verlag, 1997

6 McEliece R J. A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42~44, Jet Propulsion Laboratory, 1979

7 Kilian J, Leighton T. Fair cryptosystem, revisited. In: Advances in Cryptology, Proceedings of Crypto'95. Lecture Notes in Computer Science 963, Springer Verlag, 1995. 208~221


# Ajtai-Dwork 密码系统的误用


赵柱

(张掖师范高等专科学校计算机科学系 张掖 734000)



**摘要** Ajtai 和 Dwork 构造了一种概率公用密钥体系.这种密码系统的安全性建立在一种格问题复杂性的最坏情形上.该文的结果证明这种密码系统是很容易被误用的.如果这种系统被用于广泛使用的 key-escrow 体系中(特别是美国的一些体系中),密码系统的终端用户就可以利用这种误用来传送一些非法信息,而不必担心安全机构通过构造用户的密钥来破译这些非法信息.同样地,这种密码系统的终端用户也必须相信密码系统的制造商,因为非法制造商制造的加密或解密系统在用户一无所知的情况下,可利用这种误用把用户的密钥泄漏出去.



**关键词** 阈下通道,密码系统的误用,key escrow.

**中图法分类号** TP309