一种基于 GF(2^3)的(K,N)有意义无扩张图像分存方案

欧阳显斌 邵利平
(陕西师范大学计算机科学学院 西安 710119)

摘 要 传统有意义图像分存存在像素扩张,通常只对分存信息以较短的认证信息进行甄别,从而导致重构的秘密像素真实性无法准确鉴别。针对此问题,提出一种基于 GF(2^3)的(K,N)有意义无扩张图像分存方案。在该方案中,首先生成加密映射表并利用秘密像素的位置信息对秘密像素进行加密;然后将秘密像素的认证信息和加密像素在 GF(2^3)有限域下进行(K,N)分存,嵌入到掩体图像对应的像素中;最后将映射表的生成密钥进行(K,N)分存,计算每个子密钥的 MD5 值并公布到第 3 方公信方以防止掩体图像持有者作弊。实验结果表明,所提方案能准确地识别出秘密图像攻击区域,不存在任何像素扩张,掩体图像与秘密图像等大且嵌入分存信息的掩体图像具有较好的视觉质量。

关键词 图像分存,有意义图像分存,(K,N)门限方案,GF 有限域,无扩张
中图法分类号 TP309.7 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.054

Meaningful (K,N) Free Expansion Image Sharing Scheme Based on GF(2^3)

OUYANG Xian-bin SHAO Li-ping
(School of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

Abstract There is pixel expansion in conventional meaningful image sharing schemes which usually use short authentication information to verify the correctness of sharing information and bring defect that the facticity of the reconstructed secret image pixels cannot be accurately identified. To address these problems, a meaningful (K,N) free expansion image sharing scheme based on GF(2^3) was proposed. In the proposed scheme, firstly a key is used to generate an encryption mapping table and then this table and secret image pi-xel location information are used to encrypt secret image pixels. Secondly (K,N)-threshold scheme based on GF(2^3) is used to share the ciphered pixels and pixels authentication information and then they are embedded into N cover images. Finally the key which is used to generate encryption mapping table is also shared into N sub-keys by (K,N)-threshold scheme and the sub-keys' MD5 values are published to the third reliable party to prevent cheating from distributed cover image holders. The experimental results show the proposed scheme can accurately detect attacked regions in reconstructed secret image. By comparing with conventional methods, the proposed scheme does not have any pixel expansion and distributed cover images have better visual quality.

Keywords Image sharing, Meaningful image sharing, (K,N)-threshold scheme, Gloise field, Free expansion

1 引言

数字图像信息分存技术是图像信息安全的研究热点。现有的图像信息分存技术主要源自密码学中的秘密共享,最早由 Shamir 和 Blakley 分别结合 Lagrange 插值算法和矢量空间点的性质提出[1,2]。

文献[3-5]分别结合秘密共享方案[1,2]提出(K,N)图像信息分存方案,将秘密图像借助秘密共享算法拆分为 N 份影子图像,若至少收集到 K 份影子图像,则可完整地重构出秘密图像,否则得不到秘密图像的任何信息。文献[3-5]只是将秘密图像转换为无意义的影子图像,在公有信道传输中容易诱发攻击而使分发影子图像遭受破坏,从而降低秘密图像最终正确重构的可能性。

针对无意义图像分存方案在使用过程中存在的安全隐患,一些文献也探讨了有意义图像分存。文献[6,7]分别使用异或和恢复函数针对 2 值和灰度图像提出(N,N)有意义图像分存方案。若 N 份掩体图像中有一份遭受攻击,则秘密图像无法完整恢复,并且文献[6,7]所提策略中没有任何认证措施,参与者可对自己保管的掩体图像进行任意修改而不被发现。

相对于(N,N)有意义图像分存方案,目前使用较多的依然是基于 Shamir-(K,N)门限方案的有意义图像分存。文献

到稿日期:2014-12-22 返修日期:2015-02-10 本文受国家自然科学基金资助项目(61100239),教育部高等学校博士学科点专项科研基金资助项目(20110202120002),陕西省科技新星计划资助项目(2011kjxx17),陕西省自然科学基金资助项目(2011JQ8009),中央高校基本科研业务费支持项目(GK201402036)资助。

欧阳显斌(1992-),男,硕士生,CCF 学生会员,主要研究方向为图像分存,E-mail:xianbin5@163.com;邵利平(1978-),男,博士,副教授,硕士研究生导师,CCF 会员,主要研究方向为信息隐藏、可视分存、图像加密和稀疏表示等,E-mail:slpmaster@163.com(通信作者)。

[8]预先将秘密图像的每个像素调整到[0,251]之间,然后对其进行 Shamir-(K,N)分存,将得到的 8 位分存信息嵌入到掩体图像对应的 2×2 分块中,并调整 2×2 分块右上角位置的奇偶校验位作为认证位,由此不可避免地导致秘密图像失真,嵌入分存信息的掩体图像视觉质量下降,且认证信息只有 1 位奇偶校验位,起不到丝毫的认证作用。为避免文献[8]对秘密图像进行预先处理导致的秘密图像失真,文献[9]将 Shamir-(K,N)拓展到 GF(2^8)有限域,并通过 HMAC (Hash-based Message Authentication Code)对分存信息进行认证,但所提出的认证方法依然只有 1 位认证位,恶意的参与者依然有很大概率逃脱检验,且 GF(2^8)涉及到域上多项式环的加减乘除运算,涉及的运算代价较大。针对文献[8,9]存在的认证问题,文献[10]采用文献[3]的分存策略并用中国剩余定理生成分存信息的 4bit 认证位来进一步提升认证能力并提高掩体图像的视觉质量,然而文献[10]并不具备攻击后的修复能力。为提高修复能力,文献[11,12]使用 Lagrange 的多个系数来分存秘密图像像素和它的配对像素,使得所提方案具备一定的攻击后修复能力。

文献[8-12]为减小掩体图像膨胀和对像素的修改,一般采用较短的认证码来对分存信息进行认证,例如分别使用 1bit、4bit 或 3bit 的认证位来对分存信息进行认证,但较短的认证码也带来了较大误判概率,从而对最终重构的秘密像素的准确性无法鉴别。

为提高对秘密像素认证的准确度,文献[13]给出了结合先认证后分存的基于双变量对称多项式的认证方法,其提高了认证的准确度和掩体图像的视觉质量,但所提策略依然存在像素扩张。

为避免文献[8-12]存在的像素扩张和使用较短的认证码来对分存信息进行认证而导致最终重构的秘密像素真实性无法鉴别,本文将文献[9]的 GF(2^8)有限域运算约束为 GF(2^3)有限域运算,并借鉴文献[13]先认证后分存的思想,利用 GF(2^3)有限域下的 Lagrange 多项式的多个系数来对秘密像素和认证信息进行分存,从而相对于文献[9]降低了运算代价,提高了分存效率。所提出的策略采用 4 个认证比特位对秘密像素进行认证,从而有较高的概率保证重构出的秘密像素的真实性,且所提出的策略中秘密图像与掩体图像等大,从而相对于文献[8-13]极大减轻了分存负载。

本文第 2 节给出基于 GF(2^3)有限域的 Lagrange 分存模型;第 3 节给出基于先认证后分存的认证和嵌入机制;第 4 节给出对密钥 key 进行保护的分存和恢复机制;第 5 节给出完整的分存和恢复算法;第 6 节对所提方案进行实验验证,并与传统方法进行实验比较;最后对全文工作进行总结并给出下一步的研究方向。

2 基于 GF(2^3)有限域的 Lagrange 分存模型

文献[8,10-12]都是建立在 Shamir-(K,N)门限方案[1]的基础上,其主要思想是构建如式(1)所示的 Lagrange 多项式。式(1)中 s 是秘密,$r_1,r_2,\cdots,r_{K-1}$是随机整数,p 为素数并且满足 $s,r_1,r_2,\cdots,r_{K-1}\in[0,p)$,$N$ 为$[K,p)$范围内的整数。将 $x=1,2,\cdots,N$ 依次代入式(1),从而形成 N 个分发信息 $(1,f(1)),(2,f(2)),\cdots,(N,f(N))$。若从中任取 $t(t\geqslant K)$个不同的分发信息$(num_k,f(num_k)),k=1,2,\cdots,t$,则可按式(2)Lagrange 插值公式先对 $f(x)$进行恢复,再对分发的秘密 $s=f(0)$进行重构。式(2)中,$(num_i-num_j)_p^{-1}$ 为(num_i-num_j)在模 p 上的乘法逆元。

$$f(k)=(s+r_1k+r_2k^2+\cdots+r_{K-1}k^{K-1}) \bmod p \tag{1}$$

$$f(k)=\left(\sum_{i=1}^{t}\left(f(num_i)\prod_{j=1,j\neq i}^{t}(k-num_j)(num_i-num_j)_p^{-1}\right)\right) \bmod p \tag{2}$$

式(1)中的模数 p 只能选取素数,从而保证式(2)中任何一个模 p 下的非零值(num_i-num_j)都存在乘法逆元$(num_i-num_j)_p^{-1}$。然而在计算机中,数据通常使用二进制来存储,因此要对秘密信息进行截断处理[3,8,10]或选择大于秘密数值上界的模数来进行分存[11,12]。由此不可避免地导致嵌入的秘密信息损失一定精度或对模数空间造成较大浪费[3,8,10-12],从而降低掩体图像的视觉质量。为避免此类问题,文献[9]将 Shamir-(K,N)分存拓展到 GF(2^8)有限域,可在一定程度上缓解上述问题,但所提方法并未充分地利用 Lagrange 多项式的多个系数来尽可能地减少分存信息,同时 GF(2^8)有限域上的运算是建立在 GF(2^8)域上多项式环的基础上,涉及较大的运算代价。为避免文献[9]在 GF(2^8)有限域上进行运算所带来的高昂计算代价和提高分存效率,本文将 Shamir-(K,N)门限方案约束在 GF(2^3)有限域来缓解计算代价,同时进一步利用 GF(2^3)有限域分存多项式的多个系数分存来提高分存效率。式(3)即为 GF(2^3)有限域分存多项式。

$$f_{GF}(k)=GF(a\hat{+}b\hat{\cdot}k\hat{+}c\hat{\cdot}k^{\hat{2}}\hat{+}d\hat{\cdot}k^{\hat{3}}\hat{+}r_1\hat{\cdot}k^{\hat{4}}\hat{+}\cdots\hat{+}r_{K-4}\hat{\cdot}k^{\hat{K-1}} \bmod GF(11) \tag{3}$$

式中,a,b,c,d 为秘密值,$r_1,r_2,\cdots,r_{k-4}$为随机数,并且满足 $a,b,c,d,r_1,r_2,\cdots,r_{K-4}\in[0,8)$,$GF()$为有限域计算函数,满足如下运算性质。

性质 1 $GF(y)$将整数 y 转换为$GF(2^n)$上的 2 值多项式 $f(\dot{x})$且多项式的系数只能是 0 和 1。

例如:使用 $GF()$可将 0,1,2,3,4,5,6,7,11 分别转换为 $0,1,\dot{x},\dot{x}+1,\dot{x}^2,\dot{x}^2+1,\dot{x}^2+\dot{x},\dot{x}^2+\dot{x}+1,\dot{x}^3+\dot{x}+1$。

性质 2 $GF(a\hat{+}b)=(GF(a)+GF(b)) \bmod 2$。

性质 3 $GF(a\hat{\cdot}b)=(GF(a)GF(b)) \bmod 2$。

性质 4 $GF(a\hat{+}b\hat{\cdot}c)=(GF(a)+GF(b\hat{\cdot}c)) \bmod 2$。

性质 5 $GF(a^{\hat{k}})=GF(\overbrace{a\hat{\cdot}\cdots\hat{\cdot}a}^{k})$。

性质 6 $GF(b\hat{\cdot}a^{\hat{k}})=(GF(b)GF(a^{\hat{k}})) \bmod 2$。

性质 7 $GF(a\hat{-}b)=GF(a\hat{+}b)$。

将 $k=1,2,\cdots,N(N<8)$依次代入式(3),将得到 N 个分发信息$(1,f_{GF}(1)),(2,f_{GF}(2)),\cdots,(N,f_{GF}(N))$。同传统的 Shamir-$(K,N)$门限方案一样,从中任取 $t(t\geqslant K)$个不同的分发信息$(num_k,f_{GF}(num_k)),k=1,2,\cdots,t$,可通过式(4)先还原多项式 $f_{GF}(k)$,再通过 $f_{GF}(k)$提取出秘密信息 a,b,c 和 d。

$$f_{GF}(k)=\left(\left(\sum_{i=1}^{t}\left(f_{GF}(num_i)\prod_{j=1,j\neq i}^{t}GF(k\hat{-}num_j)(num_i\hat{-}num_j)_{GF(11)}^{-1}\right)\right) \bmod 2\right) \bmod GF(11) \tag{4}$$

式中，$(num_i \hat{-} num_j)_{GF(11)}^{-1}$为$GF(num_i \hat{-} num_j)$在本原多项式$GF(11)=\dot{x}^3+\dot{x}+1$上的乘法逆元多项式，并且满足$((GF(num_i \hat{-} num_j)(num_i \hat{-} num_j)_{GF(11)}^{-1}) \bmod 2) \bmod GF(11)=1$。

3 基于先认证后分存的认证和嵌入机制

文献[8-12]采用的认证机制都是用较短的认证码来对分存信息进行认证，即先分存后认证机制；较短的认证信息也会导致掩体图像持有者对掩体图像恶意篡改后有较大的概率逃脱检验，从而导致最终重构出的秘密图像像素的真实性难以得到检验。

与文献[8-12]不同，本文采用的认证机制是先认证再分存，即首先产生秘密像素的认证信息，然后再将秘密像素和认证信息分存嵌入到对应的掩体图像中。在这种认证机制下，一个掩体图像像素被恶意修改将会引起重构多项式发生变化，不同的多项式将有较大的概率提取出不对应的认证信息和秘密像素，而每个参与者都无法预知最终重构出的多项式，从而能有效地检测出秘密像素是否准确重构。

记秘密图像$\boldsymbol{S}=(s_{i,j})_{m\times n}$，其中每个像素$s_{i,j}$（$0\leqslant s_{i,j}<256$）可用8位2进制位串$(s_0^{i,j}s_1^{i,j}\cdots s_7^{i,j})_2$进行表示。为提高认证精度，本文采用式(5)来计算$s_{i,j}$的4比特位认证信息$check_{i,j}$，记$check_{i,j}$为$(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$，式(5)中"⊕"为异或操作。

$$check_{i,j}=(s_0^{i,j}s_1^{i,j}s_2^{i,j}s_3^{i,j})\oplus(s_4^{i,j}s_5^{i,j}s_6^{i,j}s_7^{i,j})_2 \tag{5}$$

由于数字图像相邻像素具有很强的相关性，若直接对$s_{i,j}$和$check_{i,j}$进行分存会导致秘密图像轮廓暴露的风险，因而本文采用密钥随机生成像素加密映射表对$s_{i,j}$进行加密，再进行分存。其对应的加密方法为：使用密钥key生成序列〈0,1,2,…,255〉的随机排列〈$q_0,q_1,q_2,\cdots,q_{255}$〉，根据$s_{i,j}$的值和位置信息使用式(6)进行加密得到$s'_{i,j}$，$s'_{i,j}$同样可用8位2进制位$(s'^{i,j}_0 s'^{i,j}_1\cdots s'^{i,j}_7)_2$表示。

$$s'_{i,j}=(q_{s_{i,j}}-((i+j) \bmod 256)+256) \bmod 256 \tag{6}$$

式(6)根据像素值和像素位置信息对秘密像素进行加密。相邻位置的相同像素值加密后对应为不同的像素值，从而破坏邻近像素的相关性，相对于对秘密像素直接分存，具有更高的安全性。

将$s'_{i,j}$和$check_{i,j}$依次分割成4个3比特位串来作为式(3)的系数a,b,c和d，如式(7)所示：

$$\begin{aligned} a&=(s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2\\ b&=(s'^{i,j}_5 s'^{i,j}_6 s'^{i,j}_7)_2\\ c&=(s'^{i,j}_2 s'^{i,j}_3 s'^{i,j}_4)_2\\ d&=(s'^{i,j}_8 s'^{i,j}_0 s'^{i,j}_1)_2 \end{aligned} \tag{7}$$

$s'_{i,j}$和$check_{i,j}$经式(3)分存后，可得到N份$GF(2^3)$下的分存多项式，记其对应的整数依次为$f_{GF}^{i,j}(1)$，$f_{GF}^{i,j}(2)$，…，$f_{GF}^{i,j}(N)$，$N<8$，然后将分存信息$f_{GF}^{i,j}(k)$（$k=1,2,\cdots,N$）嵌入到对应掩体图像$\boldsymbol{C}_k=(c_{i,j}^k)_{m\times n}$（$k=1,2,\cdots,N$）对应的像素中。这里可进一步搅乱掩体图像像素和秘密像素的对应关系以提高安全性。

将$f_{GF}^{i,j}(k)$嵌入到$\boldsymbol{C}_k$中的方法有2种，第1种是将分存信息$f_{GF}^{i,j}(k)$转化为3个比特位依次替换掩体像素$c_{i,j}^k$的低3位；第2种是调整掩体图像对应像素$c_{i,j}^k$的模值来嵌入分存信息，即使调整后的值$c'^k_{i,j}$满足$c'^k_{i,j} \bmod 8=f_{GF}^{i,j}(k)$，从而嵌入分存信息$f_{GF}^{i,j}(k)$。在本文中选择第2种方法进行嵌入，选择满足$c'^k_{i,j} \bmod 8=f_{GF}^{i,j}(k)$且$|c'^k_{i,j}-c^k_{i,j}|$最小的$c'^k_{i,j}$，从而使嵌入信息后的掩体图像获得较好的视觉质量。

4 对密钥key进行保护的分存和恢复机制

文献[9,12]采用HMAC来改进分存信息认证码的生成策略，但对HMAC的密钥如何管理并未涉及。这里的密钥十分重要，对其管理不当，将会带来较大的安全风险。

在本文中，密钥key用于生成加密映射表，同样至关重要，这里采用式(1)将其进行(K,N)分存，形成N个分发子密钥$f(k)$（$k=1,2,\cdots,N$），并分发给对应掩体持有者进行保管。这种策略只有不少于t个参与者提供合法子密钥才能恢复出key，因而这种策略具备更高的安全性，并将子密钥$f(k)$（$k=1,2,\cdots,N$）的MD5值公布到第3方公信方以防止子密钥持有者篡改子密钥作弊。

在恢复过程，假设有$t(t\geqslant K)$个不同的参与者提供子密钥$f(num_k)$和掩体图像$\boldsymbol{C}'_{num_k}$，$num_k\in[1,N]$（$k=1,\cdots,t$）参与秘密图像重构。首先计算$f(num_k)$对应的MD5值与第3方公信方公布的MD5是否相等，若相等，则表示$f(num_k)$合法，反之不合法。若合法的子密钥数小于门限K，将无法恢复出密钥key，从而不能对秘密像素进行解密。

为便于描述，这里假设t个参与者提供的子密钥均合法。记通过认证的参与者子密钥信息为$(num_k,f(num_k))$（$k=1,2,\cdots,t$），则可通过式(2)还原得到密钥key，并用密钥key再次生成序列〈0,1,2,…,255〉的排列〈$q_0,q_1,q_2,\cdots,q_{255}$〉用于解密像素。

对于位置(i,j)，$0\leqslant i<m$，$0\leqslant j<n$，可通过式(8)来提取掩体图像该位置嵌入的分存信息$f_{GF}^{i,j}(num_k)$。

$$f_{GF}^{i,j}(num_k)=c'^{num_k}_{i,j} \bmod 8 \tag{8}$$

将$(num_k,f_{GF}^{i,j}(num_k))$转化成有限域$GF(2^3)$下对应的多项式，按式(4)恢复式(3)，提取出式(3)的4个系数即式(7)中的a、b、c和d，得到加密后的秘密像素$s'_{i,j}=(s'^{i,j}_0 s'^{i,j}_1\cdots s'^{i,j}_7)_2$和认证码$check_{i,j}=(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$，使用式(9)将$s'_{i,j}$解密得到秘密像素$s_{i,j}=(s_0^{i,j}s_1^{i,j}\cdots s_7^{i,j})_2$，其中函数$id(v)$表示排列〈$q_0,q_1,q_2,\cdots,q_{255}$〉中$v$所对应的下标索引，即$s_{i,j}$。

$$s_{i,j}=id((s'_{i,j}+i+j) \bmod 256) \tag{9}$$

由$s_{i,j}$通过式(5)再次计算$s_{i,j}$的认证信息$(s''^{i,j}_8 s''^{i,j}_9 s''^{i,j}_{10} s''^{i,j}_{11})_2$，若$(s''^{i,j}_8 s''^{i,j}_9 s''^{i,j}_{10} s''^{i,j}_{11})_2$等于$(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$，则认为该秘密像素未被攻击且正确恢复，并置$b_{i,j}=1$，表示认证通过；反之令$b_{i,j}=0$，置$s_{i,j}=128$，即认证不通过。当所有像素处理完后即可重构出秘密图像$\boldsymbol{S}=(s_{i,j})_{m\times n}$和认证图$\boldsymbol{B}=(b_{i,j})_{m\times n}$。

5 完整的基于$GF(2^3)$的(K,N)有意义图像分存与恢复算法

结合第2节－第4节的工作，以下给出完整的基于$GF(2^3)$的(K,N)有意义图像分存和恢复算法，记为算法1和算法2。

算法1 基于$GF(2^3)$的(K,N)有意义图像分存算法

步骤1 获得秘密图像$\mathbf{S}=(s_{i,j})_{m\times n}$和N张掩体图像$\mathbf{C}_k=(c_{i,j}^k)_{m\times n}$，k=1,2,…,N，选取K和密钥key的值；

步骤2 使用式(1)将密钥key进行分存，得到N个子密钥f(k)，k=1,2,…,N，并将对应子密钥的MD5值公布到第3方公信方；

步骤 3　使用密钥 key 生成排列$\langle q_0, q_1, q_2, \cdots, q_{255}\rangle$；

步骤 4　对于$\mathbf{S}=(s_{i,j})_{m\times n}$的像素$s_{i,j}=(s_0^{i,j}s_1^{i,j}\cdots s_7^{i,j})_2$，用式(5)得到其认证信息$check_{i,j}=(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$，用式(6)得到加密后的$s'_{i,j}=(s_0'^{i,j}s_1'^{i,j}\cdots s_7'^{i,j})_2$，通过式(7)得到 a、b、c 和 d，并用式(3)进行分存，得到分存信息$f_{GF}^{i,j}(1), f_{GF}^{i,j}(2), \cdots, f_{GF}^{i,j}(N)$，$N<8$；

步骤 5　分存信息$f_{GF}^{i,j}(k)$嵌入到掩体图像$\mathbf{C}_k=(c_{i,j}^k)_{m\times n}$对应位置$c_{i,j}^k$，$k=1,2,\cdots,N$，这里可进一步搅乱掩体图像像素和秘密像素的对应关系以提高安全性；

步骤 6　反复执行步骤 4 和步骤 5，直至处理完秘密图像所有像素，得到嵌入分存信息后的掩体图像$\mathbf{C}_k'=(c'^k_{i,j})_{m\times n}$，$k=1,2,\cdots,N$，将它们和子密钥 f(k)分发给对应参与者，并销毁所有中间数据。

算法 2　基于 GF(2^3)的(K,N)有意义图像恢复算法

步骤 1　假设有 t($t\geqslant K$)个不同的参与者提供的子密钥 f(num_k)和掩体图像$\mathbf{C}'_{num_k}$参与秘密图像重构，计算 f(num_k)的 MD5 值并与第 3 方公信方的 MD5 值进行对比，若相等则认为其合法，否则不合法。若合法的子密钥数不小于门限，则继续恢复过程；否则恢复失败(这里假设 t 个参与者提供的子密钥均合法)。

步骤 2　将合法的参与者子密钥(num_k, f(num_k))($k=1,2,\cdots,t$)通过式(2)恢复得到密钥 key。

步骤 3　使用密钥 key 重新生成排列$\langle q_0, q_1, q_2, \cdots, q_{255}\rangle$；

步骤 4　对于位置(i,j)，通过式(8)提取出嵌入信息$f_{GF}^{i,j}(num_k)$，将(num_k, $f_{GF}^{i,j}(num_k)$)($k=1,2,\cdots,t$)用式(4)还原出式(3)，从而得到$s'_{i,j}=(s_0'^{i,j}s_1'^{i,j}\cdots s_7'^{i,j})_2$ 和$check_{i,j}=(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$；

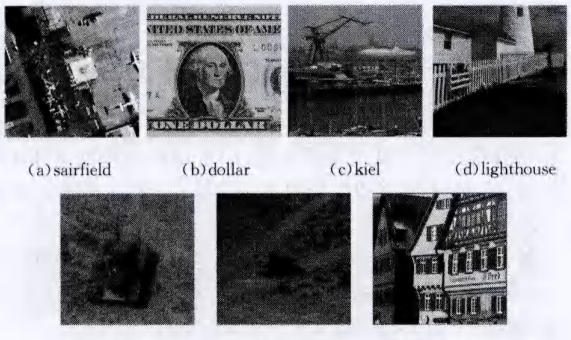
步骤 5　用式(9)将$s'_{i,j}$解密得到$s_{i,j}=(s_0^{i,j}s_1^{i,j}\cdots s_7^{i,j})_2$，并通过式(5)再次计算$s_{i,j}$的认证值$(s''^{i,j}_8 s''^{i,j}_9 s''^{i,j}_{10} s''^{i,j}_{11})_2$，若等于$(s'^{i,j}_8 s'^{i,j}_9 s'^{i,j}_{10} s'^{i,j}_{11})_2$，则置$b_{i,j}=1$，$s_{i,j}=(s_0^{i,j}s_1^{i,j}\cdots s_7^{i,j})_2$，否则令$b_{i,j}=0$，$s_{i,j}=128$；

步骤 6　反复执行步骤 4 和步骤 5，直至处理完所有像素，可得秘密图像$\mathbf{S}=(s_{i,j})_{m\times n}$和认证图$\mathbf{B}=(b_{i,j})_{m\times n}$。

6　实验

以下对本文所提策略进行实验验证，并将所提策略与文献[8-12]进行实验比较。实验测试环境为 Windows 7 操作系统，CPU 为 AMD FX(tm)-8320 8 核 CPU，单处理核心主频为 3.50GHz，内存为 16.00GB，实验编码语言为 JAVA jdk1.8.0_20。实验中采用(4,6)门限方案，即 $N=6$，$K=4$，密钥 $key=131819$。

实验测试图像分别采用分辨率为 512×512 的 8 位灰度图像 sairfield、dollar、kiel、lighthouse、tank、ruck 和 houses，如图 1(a)－图 1(g)所示，其中图 1(a)为秘密图像，图 1(b)－图 1(g)为掩体图像。



(a) sairfield　(b) dollar　(c) kiel　(d) lighthouse

(e) tank　(f) truck　(g) houses

图 1　实验测试图像

为客观评价两幅图像差异，本文使用 *MSE* 来比较两幅图像的差异，使用 *PSNR* 来衡量图像的视觉质量。*MSE* 使用式(10)计算，*PSNR* 使用式(11)计算：

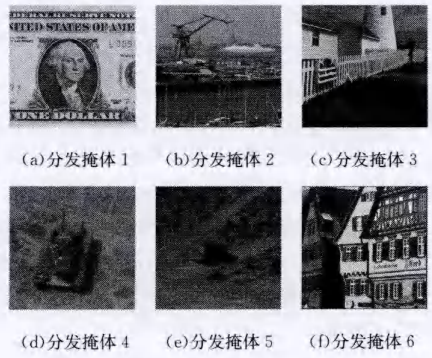
$$MSE=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(p_{ij}-p'_{ij})^2}{m\times n} \tag{10}$$

$$PSNR=10\log_{10}\frac{255^2}{MSE} \tag{11}$$

式(10)中 p_{ij}，p'_{ij} 为待比较两幅图像的像素值。

6.1　秘密图像的分存和恢复验证实验

将图 1(a)按算法 1 进行(4,6)门限分存，分存得到的掩体图像如图 2(a)－图 2(f)所示。图 2(a)－图 2(f)相对于图 1(b)－图 1(g)的 *PSNR* 分别为 40.73dB，40.68dB，40.72dB，40.74dB，40.73dB，40.56dB。参与者获得的子密钥分别为：4178813，22225831，68272871，156319931，300367009，514414103，这些子密钥对应的 MD5 分别为：0x3da8d5e47b98e303ec770cffc480a8f7，0x37a94a8f4f57caeb3e36f4c6d5a54aa4，0xf9791a2d7baf97c7f50bb28d4ae60184，0x520d2570df90b197dc7ed2cbd3335caa，0x00c107c83d336d70499c6eee1bce230a，0x1eba5935e909026ee308fa661c486b91。



(a)分发掩体 1　(b)分发掩体 2　(c)分发掩体 3

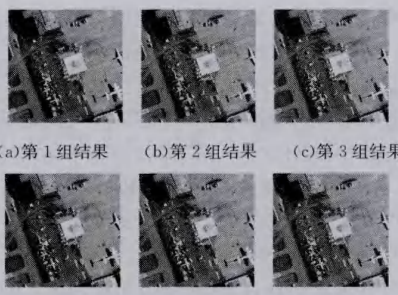
(d)分发掩体 4　(e)分发掩体 5　(f)分发掩体 6

图 2　分发掩体图像实验图样

图 2 的实验结果表明，嵌入分存信息后的掩体图像具有较高的视觉质量，攻击者难以发现嵌入的信息，并且嵌入分存信息后的掩体图像和原掩体图像等大，不存在任何像素扩张。对图 2(a)－图 2(f)按算法 2 进行恢复，其恢复参数和恢复出的秘密图像如表 1 和图 3 所示，从图 3 的实验结果可看出，在掩体图像没有遭受攻击的情况下，只有参与恢复的参与者数不小于 4 时才可无损恢复出秘密图像，否则无法恢复密钥，得不到任何秘密图像信息。

表 1　秘密图像的恢复参数

编号	参与图像	恢复秘密图像	原秘密图像	MSE
1	图 2(a)，图 2(b)，图 2(c)，图 2(d)	图 3(a)		0
2	图 2(a)，图 2(c)，图 2(d)，图 2(e)	图 3(b)		0
3	图 2(b)，图 2(c)，图 2(d)，图 2(e)	图 3(c)		0
4	图 2(a)，图 2(b)，图 2(c)，图 2(d)，图 2(e)	图 3(d)		0
5	图 2(b)，图 2(c)，图 2(d)，图 2(e)，图 2(f)	图 3(e)	图 1(a)	0
6	图 2(a)，图 2(b)，图 2(c)，图 2(d)，图 2(e)，图 2(f)	图 3(f)		0
7	图 2(a)，图 2(b)	恢复失败		—
8	图 2(a)，图 2(b)，图 2(c)	恢复失败		—



(a)第1组结果　(b)第2组结果　(c)第3组结果

(d)第4组结果　(e)第5组结果　(f)第6组结果

图3　恢复的图像

下面对本文策略的认证能力进行验证。不失一般性，以参与者1,2,3和4参与恢复为例，假设参与者1,2,3和4对应的掩体图像可能遭受图4所示的攻击。

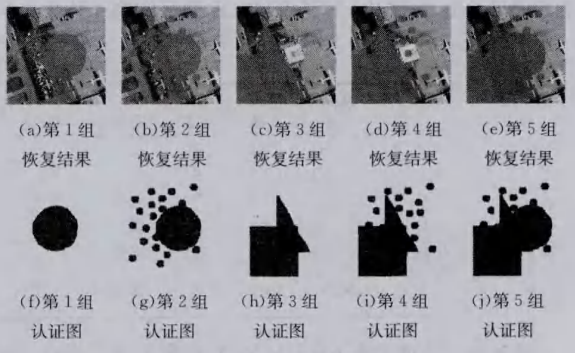
(a)被攻击后的图2(a)　(b)被攻击后的图2(b)　(c)被攻击后的图2(c)　(d)被攻击后的图2(d)

图4　被攻击后的掩体图像

对参与者1,2,3和4所提供的掩体图像按算法2进行恢复，其恢复参数和恢复出的秘密图像和认证图如表2和图5所示，从实验结果可看出本文方案的认证图可检测出所有参与者掩体图像被攻击区域的并集，其中认证图的黑色点表示认证失败的像素。从表2可看出检测出的被攻击点与实际被攻击点比率为0.935～0.938，接近于15/16，说明本文策略具有较强的认证能力。

表2　被攻击后秘密图像的恢复参数

编号	参与图像	恢复结果	原秘密图像	PSNR	认证图	攻击区域识别比率	实际攻击识别比率
1	图4(a)，图2(b)，图2(c)，图2(d)	图5(a)		17.79	图5(f)	0.820	0.935
2	图4(a)，图2(b)，图2(c)，图4(d)	图5(b)		16.14	图5(g)	0.830	0.935
3	图2(a)，图4(b)，图4(c)，图2(d)	图5(c)	图1(a)	16.34	图5(h)	0.840	0.938
4	图2(a)，图4(b)，图4(c)，图4(d)	图5(d)		14.13	图5(i)	0.847	0.937
5	图4(a)，图4(b)，图4(c)，图4(d)	图5(e)		13.74	图5(j)	0.850	0.937



(a)第1组恢复结果　(b)第2组恢复结果　(c)第3组恢复结果　(d)第4组恢复结果　(e)第5组恢复结果

(f)第1组认证图　(g)第2组认证图　(h)第3组认证图　(i)第4组认证图　(j)第5组认证图

图5　被攻击后恢复的秘密图像

6.2　与相关文献的对比

以下将本文策略和文献[8-12]进行实验比较，其对应的实验参数如图6和表3所示。实验中，为便于不同方法间的比较，作如下处理：对于文献[8-12]，由于其是将秘密图像嵌入到秘密图像4倍大小的掩体图像中，这里将图1(a)的分辨率由512×512转变为256×256，作为文献[8-12]的秘密图像。

(a1)掩体1　(b1)掩体2　(c1)掩体3　(d1)掩体4　(e1)掩体5　(f1)掩体6　(g1)恢复

(a2)掩体1　(b2)掩体2　(c2)掩体3　(d2)掩体4　(e2)掩体5　(f2)掩体6　(g2)恢复

(a3)掩体1　(b3)掩体2　(c3)掩体3　(d3)掩体4　(e3)掩体5　(f3)掩体6　(g3)恢复

(a4)掩体1　(b4)掩体2　(c4)掩体3　(d4)掩体4　(e4)掩体5　(f4)掩体6　(g4)恢复

(a5)掩体1　(b5)掩体2　(c5)掩体3　(d5)掩体4　(e5)掩体5　(f5)掩体6　(g5)恢复

图6　与同类文献对比的实验图样

表 3　与同类文献对比的实验参数

编号	相关文献	秘密图像	掩体图像	分发掩体图像	恢复图像
1	文献[8]	缩小图 1(a)	图 1(b)—图 1(g)	图 6(a1)—图 6(f1)	图 6(g1)
2	文献[9]	缩小图 1(a)	图 1(b)—图 1(g)	图 6(a2)—图 6(f2)	图 6(g2)
3	文献[10]	缩小图 1(a)	图 1(b)—图 1(g)	图 6(a3)—图 6(f3)	图 6(g3)
4	文献[11]	缩小图 1(a)	图 1(b)—图 1(g)	图 6(a4)—图 6(f4)	图 6(g4)
5	文献[12]	缩小图 1(a)	图 1(b)—图 1(g)	图 6(a5)—图 6(f5)	图 6(g5)

文献[8-12]与本文策略的性能对比如表 4 所列。从表 4 和图 6 可看出，文献[8-12]是将秘密图像嵌入到 4 倍大小的掩体图像中，且文献[8]是有损恢复秘密图像。而本文策略是无损恢复，嵌入分存信息后的掩体图像的平均视觉质量为 40.70dB，优于文献[8]的 39.15dB，虽然低于文献[9-12]的平均视觉质量，但所提策略是将分存信息嵌入到同秘密图像等大的掩体图像中，相比之前工作，本文策略不存在任何像素扩张，且嵌入分存信息的掩体图像具有较好的视觉质量。而且从表 4 中可看出，相比文献[8-12]，本文采用的认证机制是先认证再分存，是直接对恢复的秘密像素进行认证，而不是对秘密像素的分存值进行认证，从而可以对秘密像素的真实性进行准确鉴别；而文献[8-12]采用的是先分存后认证，由于对分存信息的认证存在较大的误判概率，从而对最终重构的秘密像素准确性无法鉴别。

表 4　相关文献性能对比结果

策略	方案类型	像素比率	无损恢复	适用范围	认证方式	分发掩体图像的 PSNR(单位：dB)						
						dollar	kiel	lighthouse	tank	truck	houses	平均
文献[8]	(K,N)	1∶4	否	灰度	先分存再认证	39.21	39.15	39.15	39.10	39.10	39.18	39.15
文献[9]	(K,N)	1∶4	是	灰度		41.60	41.53	41.49	41.34	41.31	41.29	41.38
文献[10]	(K,N)	1∶4	是	灰度		42.56	42.33	42.16	41.77	41.87	41.20	41.98
文献[11]	(K,N)	1∶4	是	灰度		45.13	45.08	45.14	45.12	45.10	44.98	45.09
文献[12]	(K,N)	1∶4	是	灰度		45.12	45.11	45.14	45.15	45.12	44.96	45.10
本文	(K,N)	1∶1	是	灰度	先认证再分存	40.73	40.68	40.72	40.74	40.73	40.56	40.70

结束语　针对文献[6,7]是(N,N)门限方案且没有任何认证措施以及文献[8-12]的像素扩张和用较短的认证码来对分存信息进行认证存在的安全隐患，本文给出了一种基于 GF(2^3)的(K,N)有意义图像分存方案。在该方案中，使用先认证后分存的认证机制，把秘密像素和对应的认证信息作为有限域 GF(2^3)多项式的多个系数进行(K,N)分存，充分利用模数空间和多项式系数，使得本文策略不存在像素扩张且具有较好的视觉质量。本文使用先认证后分存的认证机制，使得参与者恶意篡改的掩体图像的每个像素有接近 15/16 的概率被检测，具有较高的安全性。所提方案虽然能检测到被攻击区域，但不能对被攻击的区域进行修复。这将是下一步研究工作中重点要解决的问题。

参 考 文 献

[1] Shamir A. How to share a secret[J]. Communications of the Association for Computing Machinery, 1979, 22(11): 612-613

[2] Blakley G R. Safeguarding cryptographic keys[C]// Proceedings of 1979 National Computer Conference. New York, USA: AFIPS, 1979: 313-317

[3] Thien C C, Lin J C. Secret image sharing [J]. Computers & Graphics, 2002, 26(5): 765-770

[4] Chen C C, Fu W Y. A Geometry-based secret image sharing approach [J]. Journal of Information Science and Engineering, 2008, 24(5): 1567-1577

[5] Tso H K. Sharing secret images using Blakley's concept[J]. Optical Engineering. 2008, 47(7): 0770011-0770013

[6] 吴小天，孙伟. 基于误差扩散的图像分存方案[J]. 计算机应用，2011，31(1)：74-77
Wu Xiao-tian, Sun Wei. Image sharing scheme based on error diffusion[J]. Journal of Computer Applications, 2011, 31(1): 74-77

[7] 欧锻灏，吴小天，孙伟，等. 基于恢复函数和误差扩散的灰度图像分存方案[J]. 计算机科学，2013，40(2)：112-116
Ou Duan-hao, Wu Xiao-tian, Sun Wei, et al. Secret gray-level image sharing scheme based on recovery function and error diffusion[J]. Computer Science, 2013, 40(2): 112-116

[8] Lin C C, Tsai W H. Secret image sharing with steganography and authentication[J]. The Journal of Systems and Software, 2004, 73(3): 405-414

[9] Yang C N, Chen T S, Yu K H, et al. Improvements of image sharing with steganography and authentication[J]. The Journal of Systems and Software, 2007, 80(7): 1070-1076

[10] Chang C C, Hsieh Y P, Lin C H. Sharing secrets in stego images with authentication [J]. Pattern Recognition, 2008, 41(10): 3130-3137

[11] Chang C C, Chen Y H, Wang H C. Meaningful secret sharing technique with authentication and remedy abilities[J]. Information Sciences, 2011, 181(14): 3073-3084

[12] Chen Y H, Chang C C. Image tamper detection and recovery based on dual watermarks sharing strategy[J]. Journal of Digital Information Management, 2012, 10(1): 39-49

[13] Yang C N, Ouyang J F, Harn L. Steganography and authentication in image sharing without parity bits[J]. Optics Communications, 2012, 285(7): 1725-1735