# Using A Two Dimensional Colorized Barcode Solution for Authentication in Pervasive Computing

William Claycomb
Computer Science Department
New Mexico Institute of Mining and Technology
801 Leroy Place
Socorro, New Mexico, USA
Phone: 505-897-4142
Fax: 505-835-5587
Email: billc@nmt.edu

Dongwan Shin
Computer Science Department
New Mexico Institute of Mining and Technology
801 Leroy Place
Socorro, New Mexico, USA
Phone: 505-835-6458
Fax: 505-835-5587
Email: doshin@nmt.edu

***Abstract*— The task of establishing secure communication in a wireless pervasive computing environment faces several challenges. One of the most critical is determining the authenticity of the parties wishing to communicate. An existing method of enabling authenticated key agreement involves using a two-dimensional visual code system with digital camera enabled devices as a way of demonstratively identifying entities. This paper seeks to extend that solution by colorizing the code system, thereby extending the amount of data which can be carried by a visual tag, and thus reducing the necessary size of such tags. This will create images more easily recognized by the devices establishing secure communication in wireless pervasive computing environments. A detailed explanation of an implementation of this method is presented, along with performance evaluation and analysis.**

## I. INTRODUCTION

Communication between two or more mobile devices in a wireless pervasive computing environment can be established using a variety of methods. In such an environment, often the simplest communication is between built-in wireless networking components, such as those utilizing Bluetooth or 802.11 communication protocols. While this method of establishing communication is often simple to arrange, *securing* the communication between the devices faces larger challenges, such as complicated user involvement, the potential for the use of weak shared passwords, and the the possibility of a replay attack [1], [2], [3]. Furthermore, the very nature of the wireless communication makes it intrinsically more vulnerable than wired communication in that all transmissions sent between devices can be monitored by any device within the transmission range [4].

The general method for establishing secure communication in a wireless pervasive computing environment involves two steps. The first is to verify the identity of the devices wishing to communicate, and the second is to exchange information used to generate a shared secret key. Several proposals have been put forward to deal with the issue of identity verification. Most involve using pre-shared information to establish a shared secret key. Others involve distributed key management, in which shared keys are created and managed by a larger group of devices. A third method employs the use of secure side channels, or location-limited channels [5]. These are methods of communication which use capabilities of the devices that are not as pervasive as a wireless signal. For example, using infra-red or audio transmission is considered a location limited channel. It is this third method we seek to explore, and to extend.

One particular solution, utilizing a location limited channel, involves the use of two-dimensional barcodes to exchange information between two devices [6]. This solution, called *UbiCode*, uses black and white images called "visual tags" to share data such as a network address and hash of a public key. The visual tag is displayed on one device, and captured by the other using a digital camera. Using the information contained in the visual tag, secure communication can be established.

In this paper, we present an approach to extending this particular solution with the introduction of multiple colors in the barcode image. The use of color will allow each region of the visual tag to convey more information within the same space, allowing for a significant reduction in the number of regions needed to convey the necessary information. A smaller number of regions of the visual tag should make it easier for the reading device to recognize and extract information. Additionally, we seek to include a larger amount of data in the visual tag, allowing for more advanced networking and security features to be utilized. Lastly, using color in a two dimensional visual tag raises other issues, including those of color and region recognition. Given environmental variables, such as ambient lighting and background shading, as well as device variables, such as screen brightness and camera resolution, several problems could arise in data recognition. This paper also discusses the identification of the most important of those issues, and examines the results of testing using a variety of different environmental and equipment configurations.

The remainder of this paper is organized as follows. Section II will discuss background technologies and related works. In Section III, we discuss our proposed solution, followed by Section IV, which details our implementation and testing environment. Section V consists of performance analysis and

**1-4244-0237-9/06/$20.00 ©2006 IEEE**


Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

Apple EX1020
IPR2026-00316
U.S. Patent No. 11,664,123

evaluation, and in Section VI we discuss the need for visual tags in establishing secure communication, and explore several areas of future work related to this topic. Section VII concludes this paper.

## II. BACKGROUND AND RELATED WORKS

Securing a communication channel between devices in a wireless environment is more difficult than in a typical wired environment. Not only is the communication medium less secure, but often no pre-shared information exists. A common technique for establishing secure communications between two devices is to use Public Key Infrastructure (PKI). This involves the exchanging of public keys, which can be verified to be from the intended source through the use of a digital signature and a certificate authority. Using these verified, trusted public keys, a shared secret key can be obtained using a method such as the Diffie-Hellman key agreement protocol [7].

In wireless environments, however, the existence of a certificate authority cannot be assumed, nor can it be assumed that even if both devices contain public key that the certifying authority is trusted by both. Without trusting that the public key is authentic, the presence of a public key is useless. Therefore, another method of exchanging information must be found. Unfortunately, if the source of the information cannot be authenticated, then exchanging information cannot be done using an unsecured communication channel. The device users must find another way to securely exchange information used to generate a shared secret key.

### A. Previous Works

One of the first works to address this issue was done by Stajano and Anderson. This method, known as the "Resurrecting Duckling" protocol [8], involves a wireless device being set to an initial state. The first request for communication the device receives is assumed to be from a trusted source, and any information received from that source (such as a public key) is assumed to be authentic. This solution works well for a low-risk environment, such as a home network, but is not secure enough to completely eliminate the risk of an attacker being the first to establish communication, thereby setting up a man-in-the-middle attack.

Several solutions have emerged, some of which build upon the work of "resurrecting duckling," and others which propose new solutions. These solutions deal with shared key management, specifically how to share keys among both existing and new members of ad-hoc networks. Three main groups of key management methods exist. The first uses pre-shared information, or *a priori* knowledge, to establish secure keys. The second uses distributed key management, in which either all the nodes, or a subset of the nodes, in an existing ad hoc network act as a certificate authority, collectively issuing certificates to new nodes seeking to join the network. The third group uses no a priori knowledge, nor distributed key management to manage shared keys. Rather, this group utilizes other methods of sharing information between nodes, which frequently include means of communication besides the channel on which the secure communication is to take place.

*1) Location Limited Channels:* In the absence of pre-existing shared keys, and in situations where a new network is being formed between two devices only, neither the pre-shared information model, nor the distributed key management model will work for establishing secure communication. The devices must exchange information via a channel other than that ultimately used for secure communication. The concept of using physical contact to exchange information is proposed by Stajano and Anderson [8], and is refined by Balfanz, Smetters, Stewart, and Wong, who refer to this technique as using a *location limited channel* [5]. Specifically, they note that "Location-limited channels have the property that human operators can precisely control which devices are communicating with each other."

Location limited channels do not necessarily have to include electronically based devices. The simple act of writing a password on a chalkboard, which only people who are authorized members of the network can see, can be considered using a location limited channel. Using even a weak, written password, secure shared keys can be generated among the members of the network [9].

The security of the location limited channel is paramount, and can be categorized according to two factors, authenticity and confidentiality [10]. Authenticity is knowing that the information received was sent by the desired host. Confidentiality is keeping the transmitted data from being captured by unauthorized users. Most location limited channels meet both criteria. For example, sharing a password on a chalkboard establishes authenticity (the source is known), and confidentiality (only those in the room are able to see the password). However, some location limited channels only provide one of these criteria, or neither. If neither authenticity nor confidentiality are available, or if confidentiality is the only attribute available, then a solution requiring pre-shared knowledge is required [10]. However, if the channel provides authenticity only, it is still possible to exchange information used to establish secure communication. This information cannot contain any secret information, however. Rather, it should include information such as public keys, which can then be used to establish secure communication between devices.

For usability purposes, as well as to reduce possible errors in data entry, electronically controlled channels are mainly used to exchange information for secure key generation. Such channels could include physical contact [5], infra-red communication [11], audio transmission and detection [12], or displaying images for capture by digital cameras [13], [14].

*2) Using Digital Cameras:* The use of digital cameras as a location-limited device has been presented as a possible solution for exchanging information leading to the establishment of a real world interaction channel [6], [15]. The basic idea is to encode information in an image, which is displayed on one device and captured by a digital camera attached to a second device. The use of these images, or *visual tags*, has been proposed for a variety of different applications, such as con-



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

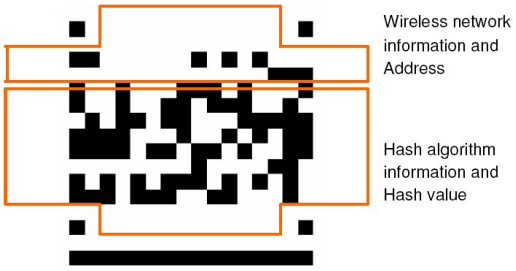



Fig. 1. The UbiCode and Data Embedded [6]



Fig. 2. A UbiColor Visual Tag

veying general information about products, allowing television viewers to capture more information about a particular story or product, asset management, or in this case, to convey network and public key information between two mobile devices [6], [16], [17], [18], [19].

The use of image capturing devices to recognize data within a barcode is not a new idea. Barcode readers have been analyzing barcodes at supermarket checkout counters for years. Specialized readers have been developed to scan two dimensional barcodes, and perform this task very quickly. However, these devices are designed for a single purpose only, to scan bar codes. They are finely tuned for that task, with attributes such as focal length being adjusted for the ideal distance between the scanner and the object being scanned [17]. Barcode scanners would not be useful in capturing digital images of objects other than barcodes.

Digital cameras attached to mobile devices, such as cell phones and PDAs, are not specifically adjusted to capture barcode images, but they can be used for this purpose. Solutions presented to date have explored the use of two dimensional black and white barcodes only. The use of color in two dimensional barcodes has not been studied carefully. In one case of a two dimensional barcode solution, the use of color is specifically avoided, citing the large difference in color values, varying lighting conditions, and the higher cost of producing colorized barcodes [15].

### B. UbiCode

Ubicode [6] is a location limited method of transferring data from one node to another, for the purpose of establishing secure communication. This method uses the display of one device and a digital camera attached to another device to transfer information via a two-dimensional barcode, as shown in Figure 1. This barcode contains two pieces of information: the network address of the host device and a hash of the host device's public key. Using this information, the guest device can establish communication (using the address included in the visual tag) and request the public key of the device using the wireless channel. The public key provided by the node at the specified address is hashed and compared to the hash received via the visual tag. If the hash values match, then the key has been verified as originating from the host device, and secure communication can be established using the public key.

Specifically, the barcode consists of 256 individual squares, of which 48 are reserved for orientation purposes. 32 squares are allocated to transmit an IP address, 128 bits are used to transmit the MD5 hash of the public key, and the remaining 48 bits are reserved for future network use. A black square indicates the presence of a 1 bit, and a white square indicates the presence of a 0 bit.

## III. UBICOLOR - EXTENDING UBICODE

We seek to extend the functionality of UbiCode. Specifically, the use of a color in the barcode is proposed. This addition should provide better image recognition, due to fewer squares being required in the tag to convey the information. Additionally, more information can be encoded in the visual tag, while still retaining a net reduction in the number of squares required to transmit the information. In reference to its roots in UbiCode, the visual tag system described here will be called *UbiColor* (see Figure 2).

### A. Colors

In UbiCode, an individual square is used to transmit one bit - a 1, and subsequently, a white square denotes a 0. This capacity can be expanded by using more than just black and white within a square. By extending the possible values to include red, yellow, green, turquoise, blue, and violet, in addition to white and black, the number of possible values the square can convey is increased to eight. By using the RGB value of the color contained within the square, a three-bit sequence can be conveyed, instead of just one bit.

By increasing the number of bits that can be conveyed with each square in the visual tag, the amount of data that can be transmitted is also increased. Using a 16x16 grid, such as the one used in UbiCode, would yield 208 bits of data. In order to transmit the same amount of data as UbiCode, UbiColor would only need 85 squares, or slightly more than a 9x9 grid of squares.

### B. Extending the Data Contained in the Code

In addition to reducing the size of the grid needed to transmit the necessary information, UbiColor also attempts to



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

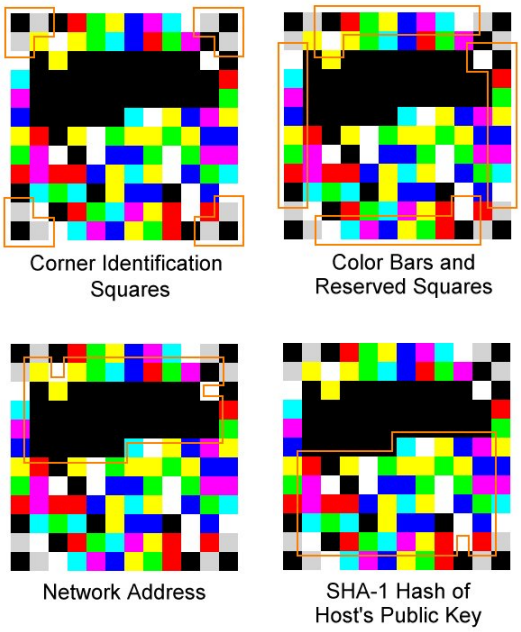



Fig. 3. UbiColor Details

extend the data included to contain more advanced information. The current implementation of UbiCode contains a 32-bit IP address, 128-bit MD5 hash of the public key, 48 bits of orientation and corner stones, and 48 bits of reserved space.

Rather than including a 32-bit IP address, UbiColor includes the capability of encoding a 128-bit address, which means the address space can include IPv6 addresses. Additionally, a SHA-1 algorithm is used to hash the host public key, requiring 160 bits of the tag to convey the information, versus 128 in UbiCode. Using a SHA-1 hash helps reduce the possibility of a successful attack on this part of the visual tag [20]. UbiColor uses just one more square than UbiCode to handle image location and orientation. These reserved squares are also used to convey color information for use as a baseline comparison by the guest node.

### C. Structure of the Visual Tag

The UbiColor visual tag is divided into four regions: tag corners, color baselines, network address, and public key hash. See Figure 3 for a detailed explanation of where these regions exist within the tag.

The tag corners are used to bound the tag, allowing the application to identify the tag within a larger image. These corners also serve as a guideline for establishing the location of each data square within the visual tag. Additionally, they determine the overall orientation and distortion of the visual tag within the image. Between the corners, along each edge of the tag, is a sequence of colors. The sequence is the same on each side, proceeding in a clockwise manner around the tag. Their purpose is to give the processing application a baseline from which to extract the color values (RGB) of each particular color within a specific image file. Given multiple environmental and device specific variables, colors will appear differently in each picture taken. By using four different color squares in four opposite ends of the image, the application can account for differences in shading within the picture as well.

Within the perimeter of the color bars is contained the data of the visual tag. The first 43 squares (counting from the top left, row by row) indicate the network information of the host device. For IPv4 addresses, only 11 color boxes are used (43 boxes * 3 bits per box = 129 bits of information). The remaining 54 squares (100 squares - 3 reserved squares - 43 network address squares) are used to indicate the 160-bit SHA-1 hash value of the host's public key.

## IV. UBICOLOR - IMPLEMENTATION

The implementation of the application to display and process UbiColor visual tags is contained in two distinct parts, each of which has both desktop and mobile versions. The first part generates and displays a visual tag on the screen of the host, then waits for a request to initiate the process of establishing secure communication. The second part processes an image file, which is a digital image of the host's device displaying the visual tag. This image file is obtained using a digital camera attached to the guest's device. Using the image, the guest's component identifies the bounds of the tag, establishes baselines for each color component, and attempts to extract information from the data portions of the visual tag.

The implementation of UbiColor was done using Microsoft Visual Studio 2005. This environment was chosen for several reasons. Most importantly, it provides outstanding support for mobile devices, using the .NET Compact Framework 2.0. Not only is mobile application development natively supported, but cryptographic namespaces exist, making functions involving both symmetric and asymmetric cryptography simple to implement. Additionally, built-in classes make generation, manipulation, and evaluation of graphical objects both easier and more intuitive than trying to use byte-level access.

### A. Test Images

The implementation was tested to determine the limits of the performance, as well as factors that affect the recognition and extraction of data from a visual tag within an image. Two sets of testing images were generated, one using a control image modified to test the limits of the algorithm in terms of brightness, rotation, distortion, size, blurriness, and noise added to the image. The second set was created using digital cameras to take pictures of visual tags displayed on different types of devices. All of the test images were processed on three different platforms, and results were collected from each test case.

To generate control images, a visual tag was generated using a known IP address and public key hash. The visual tag was saved as a bitmap image by the host application, to produce an image with distinct colors, perfect size, and no shape distortions. This image was then modified, each time producing a new image for testing. The aspects changed for each test image were as follows:



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

- Brightness
- Rotation
- Tag size within overall image
- Distortion
- Blurriness and Noise

Real photo images were tested as well. These images were taken using two digital cameras, a 1.2 mega pixel camera integrated in an HP iPAQ rx3715 PDA, and a 5.0 mega pixel Pentax Optio S50 digital camera. The images were taken of visual tags on two different displays. The first display was that of a Dell Latitude D600 laptop with a 14.1" LCD display. The second display was that of a Dell Axim X50v PDA, with a 3.7" 640x480 resolution VGA display.

Three images were captured for each combination of environmental variables, to try to eliminate errors in the image capture process, such as moving the camera or not correctly capturing the entire image. The environment in which in the photographs were taken was modified as follows:

*1) Backlight:* The overall lighting of the room was one of three values: brightly lit (using halogen lamps), dimly lit (also using halogen lamps), and brightly lit using fluorescent lighting.

*2) Background:* The surface immediately behind the test image was either a white sheet of paper (or screen), or a black sheet of paper (or screen).

*3) Resolution:* The resolution of the test cameras was set either to "high" resolution or "low" resolution. For the iPAQ camera, low resolution images were taken at 320 pixels by 240 pixels, and high resolution images were taken at 1280 pixels by 960 pixels. For the Pentax camera, low resolution images were taken at 640 pixels by 480 pixels, and high resolution images were taken at 1600 pixels by 1200 pixels.

*4) Device Brightness:* The brightness of the device display was also controlled. Images were captured from devices with their display brightness set to full brightness and half brightness. Half brightness was defined as the setting between the highest brightness and lowest brightness.

*5) Width of Tag Within Image:* Each environmental setting was captured with images containing the visual tag taking up as much of the overall image as possible, as well as images containing the visual tag taking up about half of the overall image. This measure was difficult to quantify, and was primarily at the discretion of the photographer.

### B. Test Image Samples

Figure 4 shows an image captured by the iPAQ device. The image was taken of the Dell Axim at high resolution, with the display device at full brightness, in a fluorescent-lit room, with a light background, and with the visual tag taking around half of the image width.

Figure 5 shows an image taken by the Pentax device. The image was taken of the Dell Axim, at high resolution, with the display device at full brightness, in a darkly lit room, with a light background, and with the visual tag taking as much of the overall image as possible.

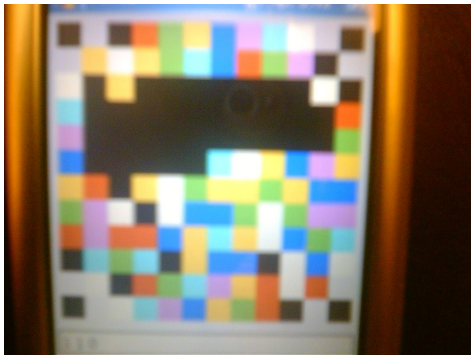


Fig. 4. Image captured by iPAQ

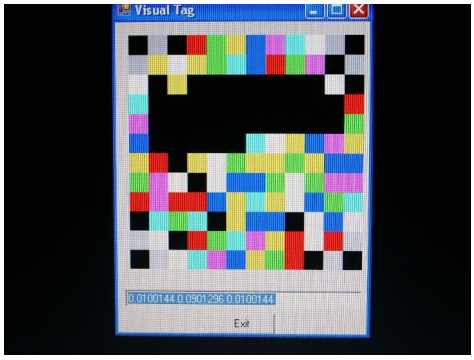



Fig. 5. Image captured by Pentax

### C. Test Platforms

Testing was conducted on three different devices. All of the images described in the previous sections (658 total images) were tested on each device, for a total of 1584 individual tests.

The testing devices consisted of the following:

- Dell Latitude D600 Laptop, with a 1.8 GHz Pentium M processor and 1.5 GB of RAM. The operating system of this device was Windows XP Professional with Service Pack 2.
- Dell Axim X50v PDA, with a 624 MHz Intel PXA270 processor and 64 MB of RAM. The operating system of this device was Microsoft Windows Mobile version 5.0.
- HP iPAQ rx3715 PDA, with a Samsung S3C2440 processor and 64 MB of RAM. The operating system of this device was Microsoft Windows Mobile 2003, Second Edition.

## V. PERFORMANCE ANALYSIS AND EVALUATION

### A. Creating and Displaying a Visual Tag

The process of creating and displaying a visual tag is fairly simple, with the most time intensive parts of the algorithm consisting of generating a public/private key pair and filling the pixels of the display bitmap.

The time to create and display a visual tag was recorded for each device a total of 100 times each. The results of these tests are shown in Table I.



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DEVICE TIMES (SEC) TO GENERATE AND DISPLAY A VISUAL TAG

| Device | Latitude | Axim | iPAQ |
|---|---|---|---|
| Minimum (sec) | 0.05 | $< 1$ | $< 1$ |
| Maximum (sec) | 0.29 | 4 | 4 |
| Average (sec) | 0.12 | 1.12 | 1.35 |

### B. Processing the Control Images

Because the control images were presented under the most ideal circumstances, the performance of each device was fairly optimal. Table II represents the minimum, maximum, and average running times for each platform processing images successfully.

TABLE II
DEVICE PERFORMANCE TIMES (SEC) ANALYZING CONTROL IMAGES

| Device | Latitude | Axim | iPAQ |
|---|---|---|---|
| Minimum (sec) | 0.16 | 1 | $< 1$ |
| Maximum (sec) | 0.98 | 5 | 3 |
| Average (sec) | 0.27 | 2 | 1 |

### C. Processing the Test Photographs

*1) Extracting Data:* A 20 second time limit was imposed on the process of finding the boundaries of a visual tag. Images that took longer than 20 seconds to identify the visual tag generally would not return correct information anyway. Table III shows the maximum, minimum, and average time for each device to process images taken with the iPAQ. Table IV shows the maximum, minimum, and average time for each device to process images taken with the Pentax. Table V shows the maximum, minimum, and average time for each device to process all images.

TABLE III
DEVICE PERFORMANCE TIMES (SEC) ANALYZING IMAGES TAKEN WITH THE IPAQ

| Device | Latitude | Axim | iPAQ |
|---|---|---|---|
| Minimum (sec) | 0.01 | 1 | $< 1$ |
| Maximum (sec) | 0.34 | 24 | 2 |
| Mean Average (sec) | 0.04 | 2.81 | 0.63 |
| Median Avgerage (sec) | 0.02 | 1 | 1 |

TABLE IV
DEVICE PERFORMANCE TIMES (SEC) ANALYZING IMAGES TAKEN WITH THE PENTAX

| Device | Latitude | Axim | iPAQ |
|---|---|---|---|
| Minimum (sec) | 0.01 | 1 | 1 |
| Maximum (sec) | 0.75 | 38 | 10 |
| Mean Average (sec) | 0.08 | 7.5 | 1.72 |
| Median Avgerage (sec) | 0.06 | 6 | 2 |

*2) Color Extraction:* When a visual tag was located, an attempt was made to extract the correct information from the data region of the tag. Table VI shows the percentage of correct data extractions.

TABLE V
DEVICE PERFORMANCE TIMES (SEC) ANALYZING ALL IMAGES

| Device | Latitude | Axim | iPAQ |
|---|---|---|---|
| Minimum (sec) | 0.01 | 1 | $< 1$ |
| Maximum (sec) | 0.75 | 38 | 10 |
| Mean Average (sec) | 0.07 | 6.00 | 1.28 |
| Median Avgerage (sec) | 0.04 | 3 | 1 |

TABLE VI
DATA EXTRACTION FROM VISUAL TAG

| | |
|---|---|
| Total number of Images Tested | 1584 |
| Number of Images with Tag Bounds Located | 1099 |
| Percent of Tag-Identified Images Returning Any Data From Tag | 90 % |
| Percent of Data-Returning Images with *Correct Data* Returned | 78 % |
| Percent of Data-Returning Images with Data Difference of 0 or 1 bit from Correct Value | 93 % |

### D. Establishing Secure Communication

Once data is extracted from the visual tag, the next step is to establish secure communication from the guest to the host. The application-specific portions of this process, such as generating a shared secret key and placing or receiving information into or out of the cryptostream buffer, were tested for performance analysis. These steps took the application under 1 second to complete, in all cases. However, the entire time to establish a secure connection between the devices ranged from 1 second to 6 seconds. No identifying feature could be determined within the application to account for the difference in time, so it was assumed to be caused by network issues between the devices. Possible factors of the network connection that could affect communication include other network traffic within the network, other network traffic on the machine, other processing occurring on the device, or disruption in the network medium, such as interference with the wireless network connection. Given that the tests occurred on isolated networks, with dedicated devices performing this task only, the most likely cause of the time discrepancy is the performance of the network medium used for communication between the devices.

## VI. DISCUSSION AND FUTURE WORKS

### A. The Case For Visual Tags

Some may question the need for a device authentication mechanism using visual tags, such as UbiColor. With secure communication solutions such as Bluetooth, why is UbiColor, with its hardware constraints, and the class of solutions it represents, necessary? There are a few answers to this question.

The first answer deals with interaction between humans and computers. Using other methods of establishing secure communications without a priori knowledge requires interaction between two humans to agree upon and enter information necessary to establish secure communication. Solutions such as UbiColor extend the functionality of establishing spontaneous secure communication to the realm of human to computer.



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

For example, a person wishing to use a PDA to transfer credit card information to a kiosk device could use a solution such as UbiColor to establish a secure link. The kiosk machine would simply need to display a UbiColor tag for the user to use in determining the network location and security characteristics of the kiosk device. No password agreement or data entry is necessary for this scenario.

Another reason for a visual tag follows this reasoning: with no password agreement or data entry necessary, the possibility of human error is reduced. Additionally, no weak points exist in the solution, such as when a weak password is used to generate a shared key. Finally, the time required to establish the communication is based on the time it takes the device to process the visual tag and establish secure communication, not on the time it takes two people to agree upon a shared password and key it into mobile devices using styli and on-screen keyboards.

However, the most direct reason for using a system of visual tags to share information is demonstrative identification [13]. A user can visually verify that the device sharing the information is the device for which communication is desired. This assures users that communication is being established between the intended devices.

### B. Future Works - Design Modifications

*1) Visual Tag Corners and Alignment:* It became apparent that the design of the visual tag was adequate, but not optimal for the tag recognition algorithm. Results of the control images showed that a tag rotated -70 degrees is actually recognized as a tag rotated +30 degrees. The addition of some orientation guidelines would help prevent this from occurring, and perhaps lead to an improved application that could identify visual tags in any degree of rotation.

Additionally, the visual tag recognition algorithm could be improved if, instead of looking for black squares, which also exist in the data portion of the tag, the corners of the visual tag were different shapes entirely. There is no requirement that two dimensional barcodes contain only squares and lines - perhaps using outlined circles as corner identifiers would improve the performance of the tag identification portion of the application.

*2) Additional Colors:* Further analysis is necessary to determine how closely the colors in the processed tags are to each other, in terms of RGB values. The colors chosen for this implementation were chosen to have RGB values as distinct as possible. If the picture quality is good enough to distinguish between color values half as far apart, the potential for addition colors would be tantalizing. Consider a color set extended to include values of 128, which is halfway between the RGB extremes of 0 and 255. This bit set would include 27 distinct values, instead of the current 8. This would more than triple the amount of data each square could convey. Even if the bit capacity of each square were doubled, using 16 colors instead of 8, the number of squares needed to carry the same amount of information would drop to 48. The visual tag data portion could be a 7x7 grid. With another layer of squares for color bar and tag boundaries, the resulting grid would be 9x9, instead of the current 12x12.

*3) Error Correction:* If additional colors were added (or even if they are not), the possibility of including error correction into UbiColor is worth consideration. The error correction could exist within the structure of the squares, where certain squares are either repeated at certain locations, or otherwise indicated through the manipulation of squared in pre-defined locations. Or, the error correction could exist within the bits the squares represent, by using a checksum or other error-correcting method to determine, and possibly repair, corrupt data. This would most likely involve using data bits for error correction, which would either reduce the number of squares available for data encoding, or require the number of squares in the entire tag to be increased. But if reliability could be improved to almost 100 percent, the sacrifice in byte capacity or resolution may be worth the cost.

### C. Future Works - Application Modifications

The application used to process the visual tag could also be modified to improve performance and add functionality to the UbiColor implementation. These modifications involve changes to optimize performance, as well as add the capability to communicate securely between two or more parties in a network.

*1) Visual Tag Identification Algorithm:* The algorithm used to identify the visual tag within the larger image could be optimized to improve visual tag recognition time. Additionally, the ability to locate smaller visual tags within the image would be an advantage to the application. However, it should not be overlooked that smaller tags within an image may not produce the distinction between colors that a larger tag would provide. A tradeoff may exist between increased tag capacity and smaller tag to image ratios that more or less cancel each other out. Determining this may require implementations of both suggestions, though.

*2) Multi-party Communication:* One very interesting area of research would be to determine a way for UbiColor to be used in multi-user configurations. UbiCode provides a means for group communication by means of an arbiter, which manages authenticating new members of the group, as well as distributing new shared public keys whenever a new member joins. UbiColor could easily use the same algorithm described by UbiCode (Figure 6), or perhaps a method could be found to include the means of encoding the information necessary to generate the shared key within the visual tag itself.

## VII. CONCLUSION

The results of this work demonstrate that the use of colorized two dimensional bar codes as a location limited channel, used to exchange information in the process of establishing secure communication, is not only feasible, but falls easily within an acceptable performance time limit. Specifically, the work detailed in this paper refutes the claim that adding color to two dimensional barcode images is not feasible for a data transfer solution [15]. Reasons given for not including



Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.

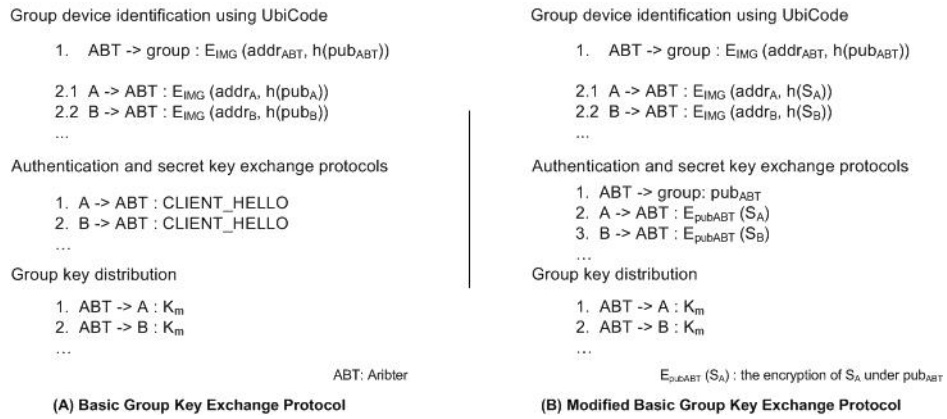



Fig. 6. Group Communication using UbiCode

color in two dimensional barcodes included the difference in color values and the uncertainty of lighting conditions. We have shown that the differences in color values can be dealt with dynamically, given the condition of the image, and that background lighting conditions do not have a large effect on the performance of the application. In general, it is the ability to locate the visual tag that limits the capability to extract data, not the ability to distinguish between colors in the image.

However, more work is needed, particularly in the percentage of images processed correctly, and especially when using lower resolution cameras to capture visual tags. Most of these improvements can be done at the application level, with modifications of color recognition thresholds and visual tag recognition algorithms. Additional changes may be necessary to the structure of the UbiColor tag itself to facilitate these improvements as well.

We also demonstrate that visual tags are a feasible method of establishing secure communication in a wireless pervasive environment. As ad hoc networking becomes more common in the future, the need for methods of establishing ad hoc secure communication will only grow [21]. Solutions such as UbiColor will enable users to create secure connections quickly and easily, opening the doors for users to access more and more security-sensitive applications, in an ever more security-aware society.

## REFERENCES


[1] T. Braun and M. Danzeisen, "Secure mobile ip communication," in *Proceedings of 26th Annual IEEE Conference on Local Computer Networks (LCN'2001)*, November 2001.

[2] P. Papadimitratos and Z. J. Haas, "Securing mobile ad hoc networks," *Handbook of Ad Hoc Wireless Networks*, 2002.

[3] G. Tuquerres, M. R. Salvador, and R. Sprenkels, "Mobile ip: Security application," April 2001.

[4] P. Yalagandula, "A survey on security issues in wireless networks," October 2000.

[5] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS02)*, San Diego, CA, February 2002.

[6] D. Shin and S. Im, "Visual device identification for security services in ad-hoc wireless networks," in *Proceedings of 20th International Symposium on Computer and Information Sciences (ISCIS'05)*, Istanbul, Turkey, October 2005.

[7] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, November 1976.

[8] A. Stajano and R. Anderson, "The ressurecting duckling: Security issues for ad hoc networks," in *Proceedings of the International Workshop on Security Protocols*, 1999.

[9] N. Asokan and P. Ginzboorg, "Key agreement in ad hoc networks," *Computer Communications*, vol. 23, November 2000.

[10] K. Hoeper and G. Gong, "Models of authentications in ad hoc networks and their related network properties," 2004.

[11] I. Pavlosoglou, R. Ramirez-Iniguez, M. S. Leeson, and R. J. Green, "A security application of the warwick optical antenna in wireless local and personal area networks," in *Proceedings of The London Communications Symposium 2002*, September 2002.

[12] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: Human-verifiable authentication based on audio," *Cryptology ePrint Archive, Report 2005/428*, 2005.

[13] J. McCune, A. Perrig, and M. Reiter, "Seeing-is-believing: using camera phones for human-verifiable authentication," in *Proceedings of The 2005 IEEE Symposium on Security and Privacy*, May 2005.

[14] F.-L. Wong and F. Stajano, "Multi-channel protocols," *Proceedings of Security Protocols Workshop*, 2005.

[15] M. Rohs, "Real-world interaction with camera-phones," in *Ubiquitous Computing Systems: Second International Symposium, UCS 2004*, Tokyo, Japan, September 2004.

[16] W. R. Claycomb and D. Shin, "Secure real world interaction using mobile devices," in *Proceedings of the Pervasive Mobile Interaction Devices (PERMID) 2006 Workshop*, Dublin, Ireland, May 2006.

[17] J. Rekimoto and Y. Ayatsuka, "Cybercode: Designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*, Elsinore, Denmark, February 2000.

[18] M. Rohs and B. Gfeller, "Using camera-equipped mobile phones," in *Advances in PervasiveComputing*, April 2004.

[19] M. Rohs and P. Zweifel, "A conceptual framework for camera phone-based interaction techniques," in *Proceedings of Pervasive Computing: Third International Conference, PERVASIVE 2005*, Munich, Germany, May 2005.

[20] X. Wang and H. Yu, "How to break md5 and other hash functions," in *Proceedings of EUROCRYPT*, 2005.

[21] A. Kaminsky, "Infrastructure for distributed applications in ad hoc networks of small mobile wireless devices," Rochester Institute of Technology, IT Lab, Tech. Rep., May 2001.




Authorized licensed use limited to: Sterne Kessler Goldstein Fox. Downloaded on April 27,2026 at 14:55:38 UTC from IEEE Xplore. Restrictions apply.