A Study of User-Friendly Hash Comparison Schemes

Hsu-Chun Hsiao*, Yue-Hsun Lin†, Ahren Studer*, Cassandra Studer*, King-Hang Wang†
Hiroaki Kikuchi‡, Adrian Perrig*, Hung-Min Sun†, and Bo-Yin Yang§
*Carnegie Mellon University, USA
†National Tsing Hua University, Taiwan
‡Tokai University, Japan
§Academia Sinica, Taiwan

***Abstract*—Several security protocols require a human to compare two hash values to ensure successful completion. When the hash values are represented as long sequences of numbers, humans may make a mistake or require significant time and patience to accurately compare the hash values. To improve usability during comparison, a number of researchers have proposed various hash representations that use words, sentences, or images rather than numbers. This is the first work to perform a comparative study of these hash comparison schemes to determine which scheme allows the fastest and most accurate comparison. To evaluate the schemes, we performed an online user study with more than 400 participants. Our findings indicate that only a small number of schemes allow quick and accurate comparison across a wide range of subjects from varying backgrounds.**

***Keywords*-Security; Human factors**

I. Introduction

Users often want secure communication while lacking any prior association. Device pairing, group key exchange/creation, and communication with remote systems (e.g., SSH or websites using self-signed certificates with HTTPS) are a few example scenarios. Without a PKI or other trusted third party available, a range of protocols are used to exchange and verify public keys or securely compute a shared key [1]–[10]. However, to ensure that a malicious entity has not compromised the key(s), i.e., performed a man-in-the-middle attack or intercepted a shared key, many protocols require the user(s) to verify that the different devices received the same messages during the protocol. Rather than comparing potentially kilobytes of data, users compare a hash or a representation of the hash of the data. Most works assume that users can perform this comparison accurately and base security guarantees on the length of the hash output or the hash representation. However, users make errors when performing such comparisons [11]. An error during this comparison renders the underlying protocol insecure.

Given humans' inability to compare long sequences of numbers accurately, researchers have proposed a number of different hash representations that (hopefully) improve usability. Rather than comparing numbers, schemes allow users to compare words [4], sentences [6], or images [3], [7], [12]. In this work, we also propose schemes where users compare Chinese, Japanese, or Korean characters, which may provide improved usability depending on a user's background. Each of these comparison schemes has strengths and weaknesses with respect to human speed and accuracy during comparison, quantifiable entropy (and thus probability of undetected attacks when users make correct comparisons), and computation overhead. The latter properties are quantified and well studied in previous works which allow protocol designers to select a hash comparison scheme that provides strong security with limited computation. However, no study has examined which approach provides the best accuracy—ensuring secure communication—with the least amount of comparison time—ensuring reduced user annoyance. The goal of this work is to conduct a user study to determine what hash comparison scheme provides the best accuracy and shortest comparison time for various users with different abilities. Based on these results, we can make an informed decision about which hash comparison scheme provides the best balance across all properties.

II. Related Work on Hash Comparison

Researchers have proposed ASCII [13][1], text [4], [6], or visual [3], [7], [12] representations of hash values. ASCII (or Hexadecimal) is natural for expressing information on computer systems but difficult for humans to quickly and accurately compare [11]. To increase usability, the text-based schemes generate ASCII representations with human recognizable structure (i.e., English words or sentences). The visual-based schemes convert hashes into images, in which humans can easily detect the differences. Fig. 1(a) to 1(i) show examples of each hash representation scheme we study in this work.

Hexadecimal digits have long been used for hash comparison because truncating a high entropy hash to a short sequence of digits (0-9) and letters (A-F) is computationally efficient. However, humans trying to quickly compare digits often make mistakes (e.g., confuse an `8` for a `0`) [11]. **Base32** utilizes a subset of digits ('2' to '7') and capital letters ('A'

[1]Recently OpenSSH 5.1 released an experimental component called ASCII visualisation, which "render SSH host keys in a visual form that is amenable to easy recall and rejection of changed host keys" [10]. Given these representations are meant to be remembered, rather than compared side-by-side, we do not consider this scheme in our study.

to ‘Z’). Hence, Base32 avoids confusion of similar-looking hexadecimal digits and increases the amount of entropy (5 bits per symbol versus 4 bits per hexadecimal digit) [13]. However, hexadecimal digits and Base 32 represent hashes as a sequence of unrelated units, which one would think, hinders humans from quickly reading and comparing the values.

The Unmanaged Internet Architecture (UIA) system represents hashes with a sequence of **English words** like “meals-abut-yuck”, in which each word (unit) is selected from a fixed size dictionary [4]. With a larger dictionary, UIA can encode hashes with fewer words. However, a large dictionary often has to contain similar words, such as “clam” and “calm”, which are difficult to quickly distinguish.

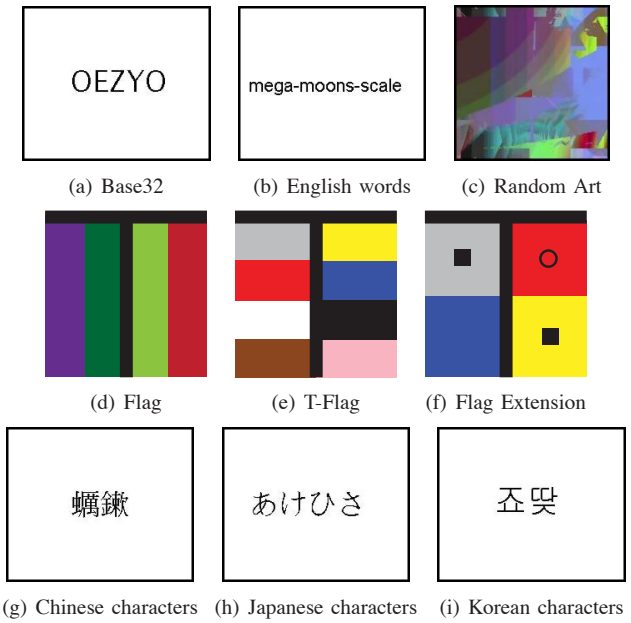
Researchers consider graphic-based representation a promising alternative for hash comparison because humans are good at quickly detecting differences in images. Perrig et al. [12] propose **Random Art** to visually represent hashes. Nevertheless, generating Random Art is computationally expensive (around 10 seconds to visualize a 160-bit hash on a handhold device) because each pixel in a Random Art image is determined by evaluating a complex arithmetic expression. Moreover, the resulting image contains an unknown amount of entropy, which may weaken security arguments in protocols which leverage Random Art [2].

Ellison et al. [3] propose **Flag**, a visual hash representation consisting of four colored strips. With 2^n possible colors in each strip, Flag contains $4n$ bits of entropy. However, Flag can only contain limited entropy because humans can have difficulty distinguishing minor variations in colors. Flag also lacks a visual cue to help users quickly determine the proper orientation when comparing images across mobile devices which may be rotated during comparison. **T-Flag** [7], which consists of 8 blocks of 8 possible colorblind proof colors with an embedded visual cue, contains 24 bits of entropy. However, Flag and T-Flag mainly express hashes by colors, rather than shapes which can encode a large amount of entropy (e.g., as Random Art does).

III. Proposed Hash Comparison Schemes

In addition to studying previously proposed hash comparison schemes, we propose and study four new schemes: **Flag Extension** and three Asian-character-based schemes, i.e., **Chinese characters**, **Japanese characters**, and **Korean characters**, as shown in Fig. 1(f)-1(i).

The Flag Extension aims to improve T-Flag by reducing the number of blocks while adding shapes on top of each block. A Flag Extension image contains 4 colored blocks (two rows and two columns). Each block has 8 possible colors (using the same colorblind proof palette as in T-Flag) and 8 possible of shapes: ‘#’,‘○’,‘I’,‘■’,‘×’, ‘∨’, ‘►’, and no shape.



(a) Base32 (b) English words (c) Random Art

(d) Flag (e) T-Flag (f) Flag Extension

(g) Chinese characters (h) Japanese characters (i) Korean characters

Figure 1. Example representations generated by the different schemes.

Chinese characters, Japanese characters, and Korean characters contain higher entropy per character compared to ASCII, and we hypothesize that people who recognize these characters are able to quickly compare characters. These schemes can be used to pair two very simple devices without full displays, e.g., LED displays designed to show single-line texts, because they only require a terminal-like non-color display with the supporting codecs (e.g., Unicode), which are often available on commodity devices.

Our Chinese characters scheme contains 9810 commonly used traditional and simplified characters from International Ideographs Core (IICORE) [14]. This is only a subset of the CJK Unified Ideograph Block (which itself covers 20,000 characters), reduced to fit in memory-limited devices like PDAs and mobile phones. In the Japanese characters scheme, we use Hiragana, the Japanese phonetic alphabet, which contains roughly 6 bits of entropy per unit. Our Korean characters scheme uses Hangul[2] (the Korean character set), which contains roughly 13 bits of entropy per unit.

IV. Study Design

In this study, we examine the performance of each hash comparison scheme with respect to accuracy rate and response time. We are also interested in knowing if participant recruitment method, gender, age, and/or language ability affects the performance. To collect such data, we built an online survey where participants can conduct a series of hash

[2]A Hangul character consists of one of 19 initial jamo, one of 21 medial jamo combinations, and *optionally* one of 27 concluding jamo combinations. To generate a unique Hangul character, we select a jamo, a medial jamo, and/or a concluding jamo, which results in a cardinality of $19 \times 21 \times 28 = 11172$. However, many of the combinations are meaningless.

comparisons and submit their background information. To recruit participants we advertised on an online service and sent emails to university classes in the US, Japan, Korea, and Taiwan.

In this section, we first present our design goals and a set of questions to be answered by this study. We then explain how to generate hash representations and specify the sampling space for a fair comparison (e.g., similar amount of entropy) between schemes. Also, we describe the procedure of our online survey. Finally, we summarize participant demographics, with details on how we recruited participants and the gender, age, and language ability, of all 436 participants.

Particularly, we study 9 hash comparison schemes: Base32, English words, Random Art, Flag, T-Flag, Flag Extension, Chinese characters, Japanese characters, and Korean characters. We do not consider hexadecimal digits because that scheme is similar to Base32 and known to be error-prone.

A. Design Goals

This study aims to answer the following questions:

- Does the participant recruitment method impact accuracy or comparison time?
- Does age or gender impact accuracy or comparison time?
- Does knowledge of a language impact accuracy or comparison time?
- What scheme(s) provide the highest accuracy?
- What scheme(s) provide the quickest response time?

We consider the problem of comparing "easy" or "hard" pairs of hash representations for each scheme. An **easy pair** consists of two representations that are either identical or apparently different. A **hard pair**, or **similar pair**, consists of two representations which are designed to be similar (but with subtle differences). Ideally, the probability of encountering a hard pair should be much lower than the probability of encountering an easy pair. In practice, however, the probabilities depend on the implementation of a hash comparison scheme and also on how humans perceive images. Hence, we separate the analysis of easy pairs and hard pairs. Easy pairs represent a baseline performance for each scheme while the hard pairs represent a worst case scenario.

B. Designing Comparison Pairs

To achieve a fair comparison, ideally each hash representation scheme should contain the same amount of information (entropy), because it is more difficult to distinguish between two slightly different representations with higher entropy. However, some schemes only allow the adjustment of the entropy in fixed intervals. For example, a Base 32 character contains 5 bits of information, so Base32 contains $5n$ bits when used with n characters. Without a way to have equal entropy, while preserving each scheme's properties, we design each scheme to carry 22 to 28 bits of information. Table I summarizes the amount of entropy in each hash representation in our study.

In the remainder of this subsection, we describe the sampling space and derive the entropy of each hash comparison scheme. We also explain the strategy to generate hard pairs, i.e., select similar-but-distinct representations from each hash representation's sampling space.

Base32. A Base32 item consists of 5 characters, with 32 possible values for each character, for a total of 25 bits of entropy. We create a similar pair by either of the following two methods: (1) creating another sequence by swapping two adjacent characters in the original sequence, e.g., VILXX and VLIXX; (2) creating another sequence by replacing a character with a very similar character, e.g., (5 ↔ S), (O ↔ Q), (2 ↔ Z). For example, PCSRA and PC5RA.

English words. The English words scheme, which consists of three words selected from a 512-word dictionary, provides 27 bits of entropy. We construct a similar pair by replacing one of the three words with a similar word, which is generated by (1) transposing two adjacent letters in a word to morph the word into another word (e.g., 'blub' and 'bulb)'; (2) selecting a word which differs by only one letter (e.g., 'moons' and 'moans'). 'house-moons-food' and 'house-moans-food' is one example of a hard pair of English words.

Random Art. The Random art image generator [12] takes any length of input and processes it with the SHA-1 hash. Theoretically speaking, it has at most 160 bits of entropy. However, there is no guarantee that Random Art is collision-resilient. Our analysis shows that with 91.4% probability a Random Art image contains 19.71 to 23.71 bits of entropy (see Appendix A for details). We use the PerceptualDiff tool [15] to measure the perceptual differences between two random art images. After generating 2000 Random Art images, we selected the five pairs with the least perceptual difference as the hard/similar pairs.

Flag. We modify Flag to output 24 bits of entropy (6 bits in each of the 4 color strips), and have a visual cue to help users determine the proper orientation of mobile devices during image comparison. Without a visual cue, an image with red-blue-green-yellow strips looks the same as another image with yellow-green-blue-red strips rotated by 180 degrees. We use 64 colors with each RGB intensity assigned one of four uniformly selected values (e.g., with intensities ranging from 0-255 we would use 0, 85, 170, and 255). To create a similar image, we copy an image and increase (or decrease) the intensity of one color of one strip by one level.

T-Flag. T-Flag gives 24 bits of entropy (3 bits in each of the 8 colored blocks). Each block is assigned a color out of 8 red-green colorblind proof colors, i.e., Black, Gray, White, Yellow, Light Pink, Red, Blue, and Brown. To create a similar image, we copy an image and swap the colors of two adjacent blocks.

Flag Extension. A Flag Extension image contains 24 bits of entropy, where each of the 4 blocks contributes 6 bits (3 bits from color and 3 bits from shape). We generate a similar image by swapping the shapes or colors of two adjacent blocks.

Chinese characters. To represent a hash by Chinese characters, two characters are selected from a set of 9810 characters. Hence, a Chinese representation contains $2\log_2(9810) = 26.52$ bits of entropy. To create a similar representation, we replace one of the two characters by a character that differs by one or two strokes, or by their radical.[3] For example, '我' ↔ '找' or '甲' ↔ '申' are only different by one or two strokes, and '游' ↔ '遊' or '獲' ↔ '穫' are different by one radical.

Japanese characters. The Japanese character scheme is composed of four Hiragana and has $4\log_2(70) = 24.52$ bits of entropy. A similar pair is generated by modifying a dakuten or handakuten (the upper-right quotation mark or circle) in one of the four characters (e.g., 'き' ↔ 'ぎ'), or by selecting two very similar Hiragana characters (e.g., 'ぬ' ↔ 'め' or 'は' ↔ 'ほ').

Korean characters. The Korean character scheme, represented by two Hangul, gives $2\log_2(11172) = 26.90$ bits entropy. To generate a similar pair, we replace one jamo with a very similar jamo, (e.g., '달' ↔ '말'or '현' ↔ '한').

C. Online Study

We performed an online user study to compare the accuracy and time needed for each of the hash comparison schemes. When participants visited our website, they completed two main steps: fill in background information and perform 27 hash comparisons. In Section IV-D, we discuss how we recruited participants and participant demographics.

Step 1. Fill in background form Participants were first asked to report their gender, age group, and language abilities. Specifically, we asked if the participant was able to recognize Chinese characters, Japanese characters, and Korean characters.

Step 2. Compare 27 pairs of hash representations After collecting demographic information, participants compared 27 pairs of hash representations (i.e., 3 pairs from each of the nine schemes). The participant were instructed to compare two hash representations at a time and decide if the representations were the same or not by pressing "same" or "different" buttons on the webpage. Fig. 2(a) shows the instructions from the webpage with two example pairs; a "same" pair on the left hand side and a "different" pair on the right hand side. Fig. 2(b) shows a screenshot of a comparison.

[3]A radical is a portion of a character that serves as an index in the dictionary. For example, the radicals of '穫' is '禾'.

During the comparisons, the order of the different schemes and the pairs encountered were randomly assigned. The detailed procedure was as follows:

1) The order of the schemes was randomly assigned, with each participant seeing each scheme once. After comparing 3 pairs from one scheme, the participant encounters 3 pairs from a randomly selected not-yet-encountered scheme.
2) Each pair of representations was selected from a pre-generated pool. The webpage shows,
   a) with probability $\frac{1}{2}$, a pair of two identical representations;
   b) with probability $\frac{1}{4}$, a pair of two obviously different representations;
   c) with probability $\frac{1}{4}$, a *distinct but similar-looking* pair, a hard pair.

   To simulate different representations that may not be perfectly aligned, each pair was displayed on the webpage (Fig. 2(b)) with one of the hash representations slightly rotated by a uniformly random angle between ± 30 degrees.
3) For each question, the participant clicked the "Same" or the "Diff" button to answer how they perceived the pair (i.e., as identical or different).
4) The answer and time spent comparing the pair were recorded and the next pair was shown.

In the experiment, same and different image pairs were shown with equal probability to ensure that a user that clicks the same button for all questions achieves a 50% accuracy. For the pairs which were different, we made half similar pairs, with the other half obviously different pairs.

D. Participant Demographics

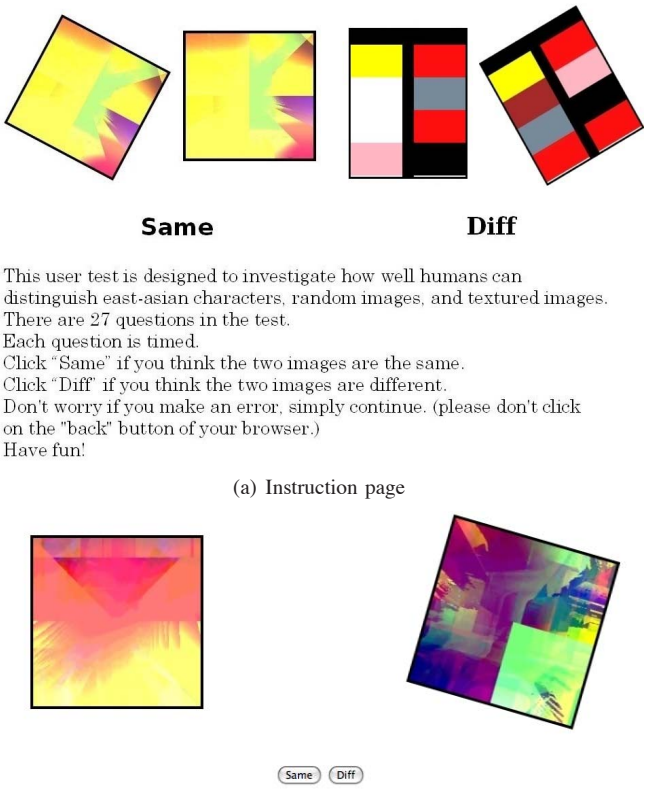
We recruited participants from two sources: 1) university classes in the US, Japan, Korean, and Taiwan and 2) Mechanical Turk (MTurk)[4] [16]. Since participants were unsupervised during the survey, participants who spent over 60 seconds for a question were dropped from the survey. There were 259 male and 177 female participants. The age of the participants varied with 239 participants 18 to 25 years old, 163 participants 26 to 40, 31 participants between 41 and 60, and only 3 participants older than 60. Information about participants' ability to recognize Asian languages can be found in Table II. Note that, some rows sum to a number greater than the number of participants from a given source because some participants can recognize more than one Asian language.

[4]Mechanical Turk is an online service that allows "requesters" to post tasks which "workers" complete in exchange for money. Most of these tasks are problems that are difficult for computers to do accurately (e.g., transcribe recordings) or require human knowledge (e.g., review a product or location).

Table I
ENTROPY IN EACH HASH COMPARISON SCHEME.

Scheme	Base32	English	Chinese	Japanese	Korean	Random Art	Flag	T-Flag	Flag Extension
Entropy (bits)	25	27	26.52	24.52	26.90	19.71-23.71*	24	24	24

*Note: The entropy of Random Art is an estimate (see Appendix A for more details).



(a) Instruction page

(b) An example of a hash comparison question

Figure 2. Screenshots from our survey website.

Table II
NUMBER OF PARTICIPANTS FROM EACH SOURCE THAT CAN RECOGNIZE THE DIFFERENT ASIAN LANGUAGES.

Source (total #)	# familiar with Chinese	# familiar with Japanese	# familiar with Korean
US (52)	13	3	3
Japan (52)	2	52	0
Korea (39)	0	1	39
Taiwan (87)	87	0	0
MTurk (212)	27	21	10

V. RESULTS AND ANALYSIS

In this section, we present the results of our study, analyze the average accuracy rate and response time for both easy and hard questions, and inspect the impact of different sources, age groups, gender, and hash comparison schemes. The statistical test ANOVA along with post-hoc contrasts were used to determine if group means were significantly different. All of our statistical tests use a significance level of 0.05. In addition, we present p-values for all statistically significant tests (i.e., $p < 0.05$). A smaller p-value shows a greater confidence in the conclusion of a statistical test that factor X has an impact on accuracy or speed. We found that

- Source, age, and gender have no significant impact on average accuracy rate across all of the schemes—excluding the Asian characters. The age group 18–25 is significantly faster than people in the 26–40 and 41–60 age groups. We also found participants from Mechanical Turk are significantly slower than Koreans on hard questions.
- Language ability affects the performance of language-based schemes (including Chinese characters, Japanese characters, Korean characters, and English words), but the influence is not always positive. In some cases, the familiarity of a language can increase the accuracy or decrease time while comparing representations which utilize that particular language. However, for English words, native speakers have lower recognition accuracy.
- In general, Base32, Random Art, T-Flag, and Flag Extension provide fast and accurate comparisons of both easy and hard questions.

In the remainder of this section, we show the result of each factor or scheme in detail.

A. Impact of Source, Age, Gender

Source. On hard questions, participants from Mechanical Turk were significantly slower than participants from Korean (on average 4.48 seconds per comparison versus 3.7 seconds per comparison, $p = 0.041$). When we ignore the Asian character schemes, there were no significant differences for accuracy or time on easy or hard items.

Age. On easy questions, age had no significant impact on the average time. On hard questions, the youngest age group was faster than both 26 to 40 year olds (on average 3.93 seconds versus 4.47 seconds, $p = 9.09 \times 10^{-3}$) and 41 to 60 year olds (on average 3.93 seconds versus 4.89 seconds, $p = 1.57 \times 10^{-2}$). There were too few people (3) to make meaningful conclusions about participants older than 60.

Gender. There was no significant difference between males and females on easy or hard pairs with respect to time or accuracy.

B. Impact of Language Ability

When selecting a hash comparison scheme, one would like to know if the ability to recognize various Asian characters will impact performance. To help answer this question, we analyzed if Asian character recognition impacted the accuracy and speed of hash comparison. Table III summarizes the average accuracy and time for participants with different language abilities on the different Asian-character-based schemes. To determine if knowledge of a language provided better accuracy or speed, we compared the performance of participants that spoke a language and participants that did not speak the language.

Table III
AVERAGE PERFORMANCE OF PARTICIPANTS ACROSS ALL OF THE ASIAN CHARACTER-BASED SCHEMES WHEN SEPARATED BY LANGUAGE ABILITY.

Language Ability	Average Easy Pair		Average Hard Pair	
	Accuracy	Time (s)	Accuracy	Time (s)
Recognize Chinese	97%	4.70	76%	4.31
No Chinese Recognition	97%	4.98	53%	5.26
Recognize Japanese	99%	4.14	65%	4.26
No Japanese Recognition	98%	4.74	55%	5.26
Recognize Korean	100%	3.72	84%	3.90
No Korean Recognition	98%	4.72	50%	5.05
English Only	92%	4.71	68%	4.60
English & ≥ 1 Asian language	96%	4.88	60%	4.64

Chinese. Knowledge of Chinese had no significant impact on speed. On easy pairs, Chinese speakers had similar accuracy to non-Chinese speakers (both 97%). However, Chinese speakers did have better accuracy on hard pairs. On hard questions, Chinese speakers had an average accuracy of 76% compared to 53% for non-Chinese speakers ($p = 3.50 \times 10^{-4}$).

Japanese. Japanese speakers were significantly faster on hard items than non-Japanese speakers (4.18 seconds versus 5.62 seconds, $p = 0.016$), but had similar speeds on easy items. Knowledge of Japanese had no statistically significant impact on accuracy. On easy questions, Japanese speakers had an average accuracy of 99% (with a 95% confidence interval of $\pm 19\%$) as opposed to 98% ($\pm 13\%$) for non-Japanese speakers. On hard questions, Japanese speakers had an average accuracy of 65% while non-Japanese speakers had an average accuracy of 55%. Despite a 10% difference in accuracy, the large variance in accuracy ($\pm 88\%$ and $\pm 94\%$ confidence intervals) on hard pairs produces a p-value of 0.179 (not statistically significant).

Korean. Korean speakers were faster than non-Korean speakers on both easy (3.72 seconds versus 4.72 seconds, $p = 0.018$) and hard Korean items (3.90 seconds versus 5.05 seconds, $p = 0.0136$). On easy questions, the accuracy was independent of knowledge of Korean. However, on hard questions Korean speakers were significantly more accurate (84% versus 50%, $p = 8.32 \times 10^{-5}$).

English. We also analyzed if English only speakers (possibly native English speakers) had an advantage on English words. Surprisingly, English only speakers were significantly **less** accurate on easy items than people who spoke Asian languages (92% accuracy versus 96% accuracy, $p = 0.018$). On hard questions, English only speakers' accuracy was not significantly different than participants that knew at least one Asian language (68% versus 60%, $p = 0.19$). Average comparison time was not affected by knowing only English.

Table IV
RESULT OF QUESTIONS FROM EACH HARD LANGUAGE-BASED PAIR.

Scheme	Question	Accuracy		Comparison
		Speaker	Nonspeaker	
Chinese characters	1	59.3%	48.4%	◯
	2	80.0%	53.1%	+
	3	76.0%	38.8%	+
	4	90.9%	69.6%	+
	5	94.4%	52.3%	+
Japanese characters	1	85.7%	66.2%	+
	2	36.4%	52.9%	◯
	3	62.5%	32.1%	+
	4	82.4%	52.3%	+
	5	83.3%	83.1%	◯
Korean characters	1	90.0%	61.9%	+
	2	100.0%	38.2%	+
	3	55.6%	21.5%	+
	4	100.0%	74.6%	+
	5	87.5%	44.6%	+
English words	1	92.6%	87.5%	◯
	2	48.0%	54.1%	◯
	3	54.5%	70.0%	◯
	4	83.8%	64.1%	+
	5	40.5%	44.2%	◯

1) Further Analysis of Hard Question Pairs: Table IV lists the accuracy of the individual hard pairs for the 4 different language-based schemes (Chinese, Japanese, and Korean characters and English words). The third and fourth columns indicate the accuracy of participants which recognize those characters versus the accuracy of participants that do not recognize those characters. For the English words, "speaker accuracy" only includes participants that only speak English. "Non-speaker accuracy" refers to participants that speak at least one of the three Asian languages. The last column indicates if speakers have a statistically significant advantage (+) or disadvantage (−) with respect to accuracy. ◯ indicates that there is no significant difference.

These results indicate that the ability to recognize a set of characters provides an advantage during at least some hard pairs (and no disadvantage). Questions that were most difficult for participants who could recognize the characters are shown in Fig. 3, including Question 1 of Chinese characters, Question 2 of Japanese characters, Question 3 of Korean characters, and Question 5 of English words. However, Question 1 of Chinese characters and Question 2

of Japanese characters were not the most difficult questions for participants who could not recognize the characters. Such a difference may be attributed to stroke order: the sequence of strokes in which a character is written. Every Asian character has a particular stroke order. A participant who has learned a set of Asian characters may compare two characters by following the stroke orders (rather than blindly trying to find a difference in two images). Hence, Asian language speakers have most difficulty comparing a pair that differs at a late stroke order, e.g., Fig. 3(a)-3(c).

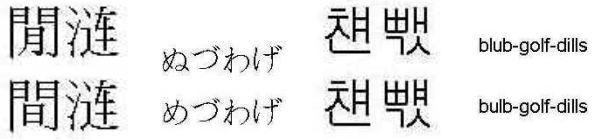
English only speakers' advantage may have been the result of which English the participants learned. English words hard pair 4 was "jowly-begs-gaol" versus " jowly-begs-goal". Those who learned the "Queen's" English may know that a "gaol" is a jail. However, few Americans may know the word. As a result, those who learned American English quickly notice the difference between "gaol" and "goal".

C. Performance of Each Hash Comparison Scheme

Table V summarizes the performance of each scheme. The first column shows the average accuracy rate over all easy pairs. The second column indicates the average accuracy over all hard pairs. The third column shows the minimum accuracy of a particular scheme among all hard pairs for that scheme. The fourth and fifth columns show the average time a user needs to compare pairs.

Our analysis shows that at least 2 schemes significantly differ in both time and accuracy for both easy and hard questions. We created partial order graphs to demonstrate the order within the schemes from most to least accurate and least time to most time (Fig. 4). In these graphs, an edge from scheme *A* to scheme *B* means participants were significantly more accurate on scheme *A* than on scheme *B* or required significantly less time.

In addition, we compared all of the schemes with Asian-language speakers removed from our sample because they perform differently in the categories Chinese characters, Japanese characters, and Korean characters based on our previous analysis on language ability. Again, the differences in accuracy and speed are statistically significant for some schemes. The partial order graphs for this set of data are shown in Fig. 5.



(a) Q1, Chinese characters (b) Q2, Japanese characters (c) Q3, Korean characters (d) Q5, English words

Figure 3. Image pairs that participants had the most difficulty comparing.

D. Suggestion

Based on our results, we suggest that protocols which require users to perform a hash comparison should use Base32, Random Art, T-Flag, or Flag Extension, because participants perform well in these schemes (for both baseline—easy pairs—and worst case—hard pairs—scenarios), independent of language ability. These suggested schemes allow faster and more accurate comparison than the other schemes independent of participants' ability to distinguish subtle color differences (i.e., Flag) or to recognize different characters (i.e., Chinese characters, Japanese characters, and Korean characters).

VI. Discussion

In this section, we discuss some topics related to hash comparison schemes that were not yet addressed in the paper: requirements, additional properties, and the limited entropy of the hash comparison schemes. In addition to accuracy and speed of comparison, hash comparison schemes also have a number of different requirements and properties that may impact what scheme works best for a given application. Hash comparison schemes provide notably less security than the underlying hash functions (20 to 30 bits of entropy versus 128 or more bits). Protocol designers must be aware of this limited entropy and design schemes that prevent an attacker from brute forcing the hash comparison scheme.

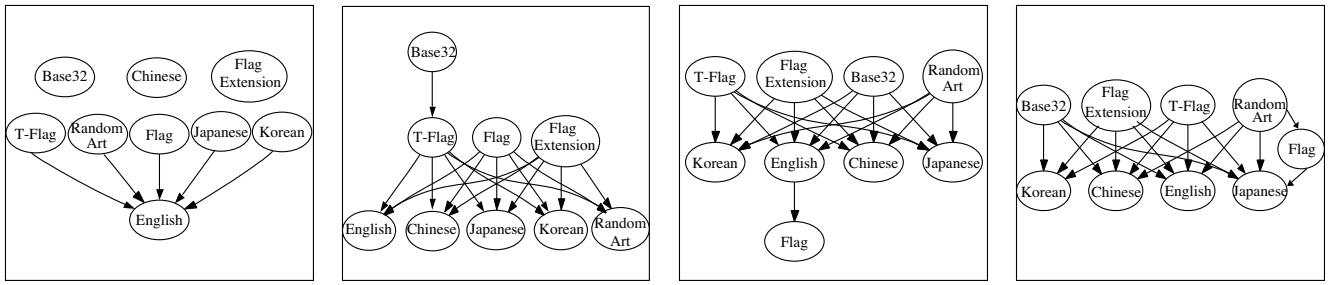
A. Additional Properties of the Hash Comparison Schemes

In this section, we discuss requirements and benefits of each hash comparison scheme.

Requirements. Some of the schemes require the devices to have certain properties to generate and display the hash comparison. Random Art, Flag, T-Flag, and Flag Extension require a *color display*. In addition, Random Art requires a high resolution display. Random Art and Flag images may appear different when printed on paper or displayed on different screens with different contrast and brightness settings. Without *duplicate display settings*, Random Art and Flag that appear different may not represent different data. The selection of 8 colorblind friendly colors for T-Flag and Flag Extension ensures that differences in contrast and brightness will not change the general colors one views. Random Art images require significant *computation power* since each pixel is computed from a complex arithmetic tree. On average, it takes about 8 seconds to generate a 180×180 pixel Random Art image on a mobile device [2], [7]. Devices that generate Chinese characters, Japanese characters, and Korean characters, require *codec support* to display corresponding character sets. Depending on the intended application and platform, some of these requirements may rule out the use of certain hash comparison schemes. For example, if the hash comparison scheme is

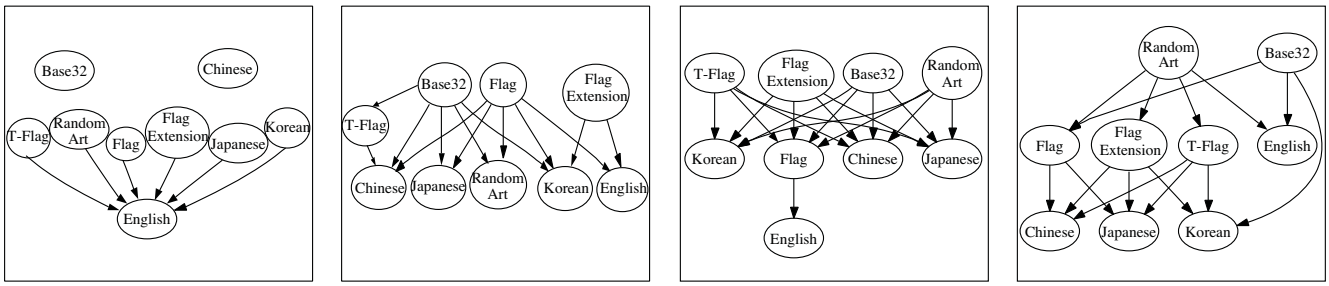
Table V
SUMMARY STATISTICS FOR EACH SCHEME.

Category	Average accuracy on easy pairs	Average accuracy on hard pairs	Minimum accuracy on a hard pair	Average time for easy pairs (sec)	Average time for hard pairs (sec)
Base32	97%	86%	71.1%	3.39	3.51
Chinese characters	97%	59%	51.4%	4.89	5.01
English words	94%	63%	42.5%	4.80	4.63
Flag Extension	98%	88%	62.7%	3.93	4.02
Japanese characters	98%	57%	39.1%	4.64	5.07
Korean characters	98%	54%	25.7%	4.61	4.92
Flag	97%	50%	15.1%	3.70	4.28
Random Art	98%	94%	60%	4.77	3.21
T-Flag	98%	85%	70.8%	3.99	4.00



(a) Accuracy (Easy Pairs Only) (b) Speed (Time Spent on Easy Pairs) (c) Accuracy (Hard Pairs Only) (d) Speed (Time Spent on Hard Pairs)

Figure 4. Partial order of schemes with all participants.



(a) Accuracy (Easy Pairs Only) (b) Speed (Time Spent on Easy Pairs) (c) Accuracy (Hard Pairs Only) (d) Speed (Time Spent on Hard Pairs)

Figure 5. Partial order of schemes when the three Asian-language speakers are excluded.

meant for computationally limited mobile devices, Random Art's computation requirement makes it a suboptimal choice. This comparison is summarized in Table VI, where a ● indicates "required".

Additional benefits. A scheme is *describable* if users can clearly describe the representation without showing it to another user (e.g., speaking the characters directly or stating the colors). It is very difficult to accurately describe some of the schemes. English words and Base32 can be easily spelled using letters and numbers (or words if the English words are common). A user will be unable to accurately describe Random Art images since some Random Art images contain a series of bars as part of a gradient of one color, but of different widths. A user can describe the colors, shapes, and order in T-Flag and Flag Extension. In Flag, minor variations in the colors may be difficult to describe without knowledge of the underlying RGB values. For Asian characters, users that speak the language can speak the given characters, but others may have trouble describing the characters. Even for random Korean characters, those familiar with the language can decompose the characters into the proper jamos.

B. Limited Entropy of the Hash Comparison Schemes

Each hash comparison scheme represents a hash h by an image or text representation $R(h)$ that outputs around 20-30 bits of entropy. To subvert most protocols, an attacker has to break the second-preimage resistance of the hash representation, as opposed to the hash (i.e., given a hash h the attacker finds some different data and hash h' such that $h \neq h'$, but $R(h) = R(h')$). Given modern devices can perform 2^{20} or more hash computations a second, it is feasible for

Table VI
SUMMARY OF REQUIREMENTS.

Scheme	Color Display	Duplicate Display Settings	Computation Power	Codec Support
Base32	-	-	-	-
English words	-	-	-	-
Random Art	●	●	●	-
Flag	●	●	-	-
T-Flag	●	-	-	-
Flag Extension	●	-	-	-
Chinese characters	-	-	-	●
Japanese characters	-	-	-	●
Korean characters	-	-	-	●

an attacker to find such a h' in a few minutes.

To prevent attacks, we could encode the entire output of a secure hash function (such as all 160 bits from SHA-1) by a hash representation. However, increasing entropy is not a solution because it sacrifices usability and accuracy. With more entropy, representations will contain longer sequences of characters or more minute details which will lead to increased time and errors during comparisons.

Rather than increasing entropy, the protocol should use commitments [17] or other techniques to ensure security. Commitments make participants' inputs to the hash representations unpredictable and prevent participants from modifying their data in response to other participants data. Without prior knowledge of the inputs and no way to change inputs after learning other participants' inputs, a malicious party is limited to a single attempt to find a collision for the hash representation. With a single guess, the chance of two hash representations with n bits of entropy being equal is 2^{n-1}.

VII. CONCLUSION

We found that Base32, Random Art, T-Flag, and Flag Extension provide the best balance across all properties—quick high accuracy comparison independent of language ability. We conclude with a decision tree (Fig. 6) that suggests proper visual comparison schemes (out of these four schemes) based on device's capability, i.e., whether a device has a high resolution display and/or sufficient computation power. Hence, these suggested schemes offer the best tradeoffs among accuracy, speed, and usability for each branch.

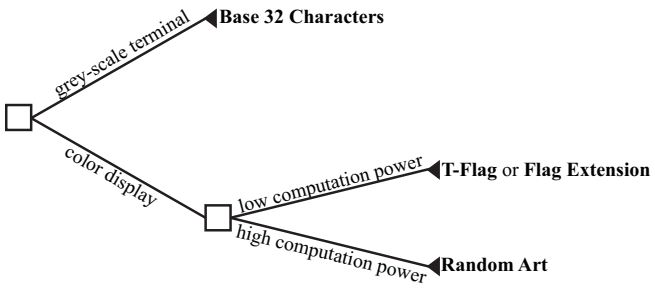


Figure 6. Hash comparison scheme selection decision tree.

ACKNOWLEDGMENT

This research was supported in part by the iCAST project, National Science Council, Taiwan under the Grant NSC96-3114-P-001-002-Y, CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and MURI W 911 NF 0710287 from the Army Research Office, grants CNS-0347807 and CCF-0424422 from the National Science Foundation. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, iCAST, NSF, or the U.S. Government or any of its agencies.

REFERENCES

[1] D. Balfanz, D. Smetters, P. Stewart, and H. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, 2002.

[2] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu, "GAnGS: Gather Authenticate 'n Group Securely," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking*, Sep. 2008.

[3] C. Ellison and S. Dohrmann, "Public-key support for group collaboration," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 4, pp. 547–565, 2003.

[4] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris, "Persistent personal names for globally connected mobile devices," in *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Nov. 2006.

[5] E. Gabber and A. Wool, "How to prove where you are: Tracking the location of customer equipment," in *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS)*, 1998.

[6] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: Human-verifiable authentication based on audio," in *International Conference on Distributed Computing (ICDCS)*, 2006, p. 10.

[7] Y.-H. Lin, A. Studer, H.-C. Hsiao, J. McCune, K.-H. Wang, M. Krohn, P.-L. Lin, A. Perrig, H.-M. Sun, and B.-Y. Yang, "SPATE: Small-group PKI-less Authenticated Trust Establishment," in *Proceedings of the 7th ACM/USENIX International Conference on Mobile Systems, Applications, and Services*, Jun. 2009.

[8] Linksky, J. et al., "Simple Pairing Whitepaper, revision v10r00," http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf, August 2006.

[9] V. Lortz, D. Roberts, B. Erdmann, F. Dawidowsky, K. Hayes, J. C. Yee, and T. Ishidoshiro, "Wi-Fi Simple Config Specification, version 1.0a," February 2006, now known as Wi-Fi Protected Setup.

[10] OpenBSD project, "OpenSSH v. 5.1," http://www.openssh.com/, Jul. 2008.

[11] E. Uzun, K. Karvonen, and N. Asokan, "Usability analysis of secure pairing methods," in *Usable Security (USEC)*, Feb. 2007.

[12] A. Perrig and D. Song, "Hash visualization: A new technique to improve real-world security," in *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC)*, July 1999, pp. 131–138.

[13] S. Josefsson, "RFC4648: The Base16, Base32, and Base64 data encodings," http://www.ietf.org/rfc/rfc4648.txt, Oct. 2006.

[14] "The unicode standard, 5.0, chapter 11," 2006.

[15] H. Lee, "Perceptual image diff," http://pdiff.sourceforge.net/, Dec 2006.

[16] Amazon Web Services, "Amazon mechanical turk (MTurk)," https://www.mturk.com/, Nov. 2005.

[17] M. Blum, "Coin flipping by telephone," in *Advances in Cryptography*, Aug. 1982, pp. 11–15.

[18] A. Orlitsky, N. Santhanam, and K. Viswanathan, "Population estimation with performance guarantees," in *IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 2026–2030.

APPENDIX

Despite that Random Art is designed to encode a large amount of entropy (e.g., 160 bits of SHA-1) [12], to the best of our knowledge, there is no theoretical analysis showing that its perceptual entropy (the amount of information perceived by humans) is close to the encoded entropy. To tackle this entropy estimation problem, we randomly sample a subset of Random Art images and count the number of perceptually similar images in the subset, by which we can statistically estimate the total number of perceptually different images.

We use PerceptualDiff [15], a tool that measures the perceptual difference between two images, to help us identify perceptually similar images in an image set. Though PerceptualDiff is not perfect — it may claim two different (or similar) images are somewhat similar (or different), it still provides a good starting point for our analysis. We generate 3709 Random Art images and evaluate the difference of each pair of images by PerceptualDiff. After manually inspecting the 50 pairs with the least perceptual differences, we found that 6 images are indeed very similar.

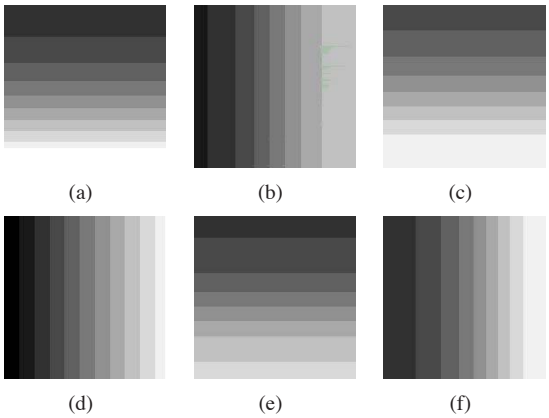


Figure 7. 6 perceptually similar images captured by PerceptualDiff. There are 4 repetitions: image (b) and (c) repeat image (a), and image (e) and (f) repeat image (d).

Orlitsky et al. [18] propose a statistic estimator $\hat{k}(N_r, r)$ to estimate a population k by random sampling, i.e.,

$$\hat{k}(N_r, r) = \frac{N_r^2}{2r} \tag{1}$$

Here N_r is the number of samples until there are r repetitions. For example, in a sequence of c, g, c, s, g, c, v, $N_1 = 3, N_2 = 5, N_3 = 6$. In the context of image sampling, we define a repetition to be an image that is perceptually similar to a previous shown image.

Equation 2 shows the confidence level of this estimator, expressing by the probability that the estimate falls outside of $[k(1-\alpha), k(1+\alpha)$.

$$\Pr(\hat{k}(N_r, r) \notin k(1 \pm \alpha)) < \frac{(1+\alpha)^r}{e^{r\alpha}}, \quad \alpha \geq 1. \tag{2}$$

The result of PerceptualDiff (together with our manual inspection) shows that there are 4 repeated instances (i.e., perceptually similar images) in the first 3709 samples. Fig. 7 shows that image (b) and (c) "repeat" image (a), and image (e) and (f) "repeat" image (d), i.e., $\alpha = 3$. We conclude that the number of perceptually different images generated by Random Art is $3709^2/4 = 3439170.25 \approx 2^{21.71}$, with only 8.6% that Random Art will carry more than 23.71 bits or less than 19.71 bits.