D-Card: A Distributed Mobile Phone Based System for Relaying Verified Friendships

Adam C. Champion, Boying Zhang, Jin Teng
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH, USA
{champion, zhangboy, tengj}@cse.ohio-state.edu

Zhimin Yang
Microsoft Corporation
One Microsoft Way
Redmond, WA, USA
yang.1070@gmail.com

***Abstract*—Cyber-physical networking systems (CPNSs) closely bridge cyberspace and the physical world. Cyberspace includes not only the Internet, but also telephone networks and short-range communications. CPNSs arise in many application domains, including social networking. Social CPNSs connect people embodied in the physical world with cyber social networking services to facilitate social interactions, including friendship formation. These services can be characterized by their dependence on Internet connections to operate. An important class of social CPNSs are mobile phone based ones. However, there is a lack of friendship verification in mobile phone based social CPNSs that allow miscreants to masquerade as friends. In this paper, we present *D-Card*, a mobile phone based social CPNS that provides friendship verification. D-Card provides an electronic name card that encodes relationship information for a person with his contact information. The name card includes a public key and digital signature. Comparing this public key with one from a trusted source enables identity verification. D-Card leverages a Bluetooth SDP toolkit to exchange information without requiring connection establishment. D-Card is a purely distributed CPNS that requires no Internet access or infrastructure. We implement the D-Card CPNS in Java ME and Bluetooth. Our experiments with real-world mobile phones illustrate its potential for friendship verification in mobile phone based social CPNSs. To the best of our knowledge, D-Card is the first such CPNS designed for this purpose.**

I. Introduction

A. Motivation

Cyber-physical networked systems (CPNSs) tightly couple cyberspace and the physical world. They have physical inputs and outputs and comprise various networked components. In general, cyberspace not only encompasses the Internet: it also includes telephone networks and short-range communications. The physical world encompasses us humans, various types of machinery, battlefields, and so on. CPNSs span many application domains, one of which is social networking.

We refer to CPNSs in this domain as *social CPNSs*. Social CPNSs bridge people in the physical world with cyber social networking services. These services help facilitate social interactions, including friendship formation. There are two main types of these services: (1) Services that require Internet connections; and (2) Services that do not require Internet connections. The following usage scenarios typify such cyber social networking services:

– *Online social networking services (Internet connection required):* Suppose Alice and Bob connect to an online social networking service like Facebook [6] from their laptops' Internet connections. They chat via the service and agree to meet at a coffee shop. Their laptops and Facebook connect them via the Internet in cyberspace, which leads to their interaction in the physical world.

– *Bluetooth/WiFi proximity (no Internet connection required):* In a student union, Clint scans for nearby Bluetooth and WiFi devices from his laptop. He discovers his friend Debra's laptop nearby. He looks around, finds Debra nearby, and talks with her. They agree to play an online game later that night. Their laptops connect them via Bluetooth and WiFi in "cyber space," which leads to their interaction in the physical world, which in turn leads to their cyberspace interaction.

There is another type of social CPNSs—mobile phone based ones. Mobile phones have rich communication features such as Internet access, cellular telephony, and short-range communication. They are highly pervasive in society. Over four billion people worldwide own mobile phones [1], [19], which always accompany their owners. Mobile phone based social CPNSs connect people as in the above scenarios. Exemplary CPNSs include Facebook as well as systems like PeopleTones [10] that notify people of nearby friends.

However, mobile phone based social CPNSs have a serious problem: they cannot verify claimed friendships. Miscreants can easily masquerade as "friends" even if contact information is supplied via physical or electronic business cards. For examples, in the first scenario, Bob might be posing as Alice's "friend" with malicious intent, and, in the second scenario, Mallory may masquerade as Debra.

B. Our Contributions

This paper presents D-Card, a mobile phone based social CPNS that enables friendship verification. D-Card stores relationship information for a person in addition to his contact information. The relationship information signifies the friendship between him and one of his friends. It includes a public key and a digital signature. The contact and relationship information are shared among social ties. Friendship verification is enabled by comparing the relationship information's public key with one

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

from a trusted source such as a personal webpage or email. To our knowledge, D-Card is the first mobile phone based social CPNS designed for such verification.

This work makes the following specific contributions:

– We design a *dedicated friendship name card* (*D-Card* for short) that cryptographically encodes the trust relationship between two friends and provides a public key and digital signature to verify it. The relationship information and signature attest to the cardholder's identity. Comparing the D-Card's public key with the cardholder's public key from a trusted source prevents an impostor from masquerading as a friend.

– We leverage our toolkit [3] to verify D-Cards via Bluetooth without establishing a connection. Our approach enables unfamiliar parties to verify identities during encounters in which connection establishment is inappropriate.

– We implement our D-Card CPNS on real-world mobile phones using Java ME and Bluetooth, which are supported by many commercial off-the-shelf mobile phones on the market. Thus our CPNS can easily run on these phones with minimal cost and configuration.

C. Typical Usage Scenarios

Two typical usage scenarios illustrate our D-Card CPNS: (1) researchers networking at a conference; and (2) business partners encountering each other at a meeting.

– *Researchers Networking at a Conference:* Suppose Ellen, Fred, and George are researchers at a conference and all of them use our D-Card system. Ellen advises her student Fred and Ellen and George are research collaborators. Ellen wants to relay her "advised student" relationship to George. She creates a "Fred" D-Card including this relationship and her public key and talks to George. Our D-Card CPNS relays this D-Card to George via Bluetooth without establishing a connection. He then verifies the relationship by comparing the public key on the D-Card with that from a trusted source such as Ellen's website. If the keys match, a window pops up on his phone notifying him of this. George can then conclude that Fred is a student who Ellen advises.

– *Business Partners' Encounter at a Meeting:* Suppose Henry, Irene, and Janet work for Company A, Company B, and Company C, respectively, and all of them are D-Card users. Their companies are working together on a project and they are at a project meeting. Henry and Irene are business partners from a previous project and Henry wants to relay his "business partner" relationship to Janet, who he encounters for the first time at the meeting. He creates an "Irene" D-Card including this relationship and his public key and talks to Janet. Our D-Card CPNS relays this D-Card to Janet via Bluetooth without establishing a connection. She then verifies the relationship by comparing the public key on the D-Card with that from a trusted source such as Henry's corporate email. If the keys match, a window pops up on his phone notifying her of this. Janet can then conclude that Irene is Henry's business partner.

These usage scenarios illustrate D-Card's potential to facilitate trustworthy social interactions among strangers. Since George has verifiable knowledge of Ellen's "advised student" relationship with Fred, George is more likely to trust this relationship than if she told him "Fred is a student who I advise" without providing any supporting evidence. Thus George is more likely to trust Fred in social interactions than if George had not verified Ellen and Fred's relationship. Similarly, since Janet has verifiable knowledge of Henry's "business partner" relationship with Irene, Janet is more likely to trust this relationship than if Henry told Janet "Irene and I are business partners" without providing any supporting evidence. Thus Janet is more likely to trust Irene in social interactions than if Janet had not verified Henry and Irene's relationship. D-Card can perform similarly in other triadic encounters in which one member of the triad has a relationship with a second member thereof and the first member wants to relay this relationship to the third member of the triad.

The rest of the paper is organized as follows. Section II reviews related work. Section III describes the entire design of the D-Card system and its implementation. We evaluate the D-Card system in Section IV. Section V concludes the paper.

II. Related Work

This section reviews prior work related to friendship verification. The closest work to ours is vCard [18], a file format standard for electronic business name cards. vCards contains information including a user's full name, address, phone numbers, photographs and other private information. They can be exchanged directly on the World Wide Web or attached to email messages. Our D-Card CPNS augments the original function of an electronic name card and adds dedicated relationship information that authenticates users.

Online social networking services such as Facebook [6], MySpace [13], and LinkedIn [11] offer users opportunities to share friendships with each other via personalized profiles. However, these services offer no friendship verification whatsoever and miscreants can easily masquerade as one's "friends" for nefarious purposes [16], [17]. In contrast, our D-Card CPNS provides cryptographic verification of friendships that prevents such masquerade.

There are several mobile phone based social networking systems that help people find friends nearby. PeopleTones [10] detects nearby friends and provides notifications of their presence. ContextContacts [14] re-designs mobile phones' contact lists to cue users to callers' locations. Hummingbird [7] uses short-range radio to provide group members continuous aural and visual indication of their proximity. Just-for-Us [9] aids friend discovery by matching users' locations or social settings with corresponding information stored in a central database. FriendZone [4] provides a suite of mobile location-based community services to assist users' discovery of each other. MobiLuck [12] lets users share their locations they can find each other and interact face-to-face. However, none of these systems provides friendship verification among strangers, which our D-Card CPNS does. Moreover, systems like Just-for-Us and MobiLuck rely on central servers that are subject to compromise and attack, leading to a central point of failure. These systems require Internet connectivity that may not always

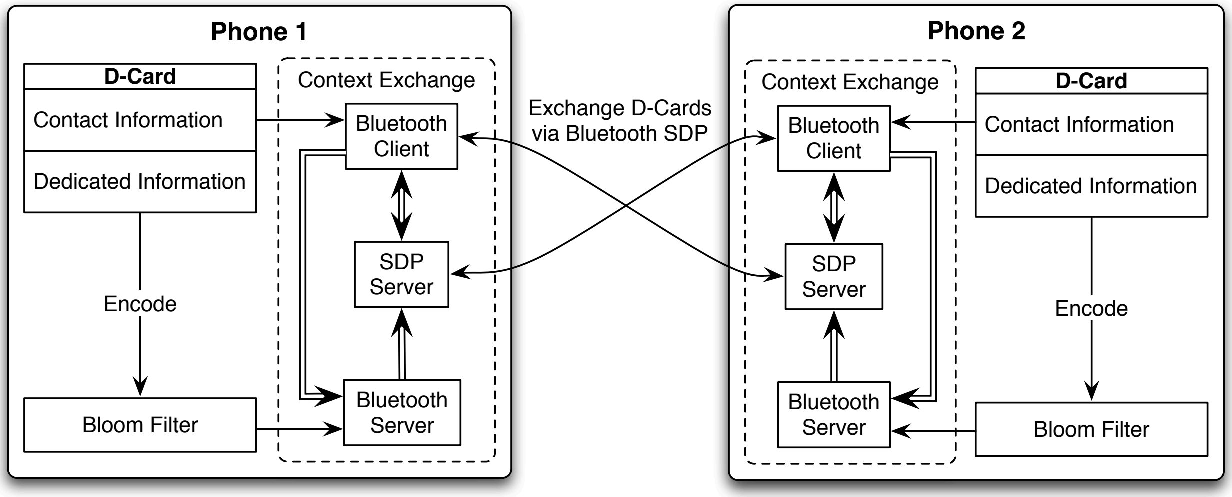


Fig. 1. D-Card System Architecture.

be available. By contrast, D-Card does not depend on central servers or Internet connectivity to function.

III. System Design and Implementation

In this section, we discuss our D-Card CPNS system design and implementation. We first describe our design considerations and then we illustrate our system architecture. Next, we discuss individual components in this architecture, namely, the D-Card and its exchange mechanism. We conclude the section with our system implementation.

A. Design Rationale

D-Card assumes that users are willing to share some sensitive information on electronic name cards. Our design rationale is guided by the following goals:

– *Verifiability:* Two mobile phone users should be able to verify friendships so they can interact with each other based on these friendships. Otherwise, people can impersonate others to steal their sensitive information.

– *User Transparency:* Users do not want to be hassled by frequent mobile phone notifications. Our system should not interrupt users until new D-Cards are discovered. The discovery procedure should be transparent to the user.

– *Privacy:* Given that our system must collect contact information stored in mobile phones to exchange D-Cards among anonymous users, we should eliminate the possibility of disclosing a user's sensitive information to anyone—either unauthorized parties or his current and future friends.

– *Power Efficiency:* Considering that mobile phones have resource constraints such as limited computing power and battery life, the D-Card discovery process should have low power overhead in order to have minimal impact on common mobile phone use.

– *Compatibility:* A verified electronic name card service should not change existing protocols. Otherwise, the mobile phone operating system might become inconsistent, which can degrade the quality of other mobile phone services.

B. System Architecture

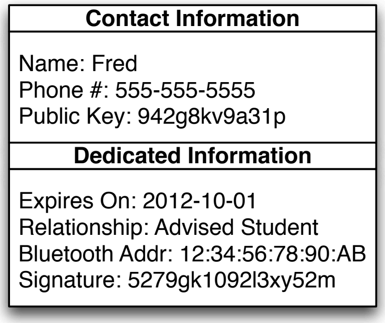
As shown in Figure 1, our system architecture consists of two major components to achieve the above design goals. The first component is the dedicated friendship name card, which provides two types of information: normal contact information and dedicated information that captures a particular relationship. The dedicated information provides a digital signature that helps the D-Card recipient verify the friendship between the D-Card sender and his contact. The second component is the D-Card exchange component. This component encodes D-Card information using a Bloom filter [2] and exchanges the filter using Bluetooth Service Discovery Protocol (SDP). This component does not change SDP, which guarantees compatibility with existing Bluetooth-enabled mobile phones.

The two components are used together to unobtrusively exchange D-Cards without requiring tedious Bluetooth connection establishment. The corresponding workflow has two stages:

– *D-Card Generation:* The D-Card system looks up names and phone numbers corresponding to a user's contacts. The user then enters his relationship with his contacts. Using his public key, our system generates D-Cards for each of them. Each D-Card includes a digital signature created using his private key. Using Bluetooth SDP, these D-Cards are stored in his mobile phone. (Users can also disseminate D-Cards via the World Wide Web or e-mail.)

– *D-Card Exchange:* When D-Card users are in Bluetooth communication range, they unobtrusively exchange their D-Cards via Bluetooth SDP. To help protect user privacy and minimize data communication, we use a Bloom filter to encode the D-Card's information in SDP. Strangers can then verify this information without having to establishing a Bluetooth connection. Specifically, they compare the D-Card's public key with the cardholder's public key from a trusted source. If verification is successful, a box pops up on their mobile phones' screens informing them of this.

Our design thus provides user privacy and friendship verifiability with the D-Card's dedicated information.



Contact Information

Name: Fred
Phone #: 555-555-5555
Public Key: 942g8kv9a31p

Dedicated Information

Expires On: 2012-10-01
Relationship: Advised Student
Bluetooth Addr: 12:34:56:78:90:AB
Signature: 5279gk1092l3xy52m

Fig. 2. D-Card Structure.

C. D-Card

In this section, we introduce the design of the dedicated friendship name card (*D-Card* for short). Unlike a vCard, which stores personal contact information in a uniform way, a D-Card stores this information and an additional piece of information that is *dedicated* to the cardholder's *specific* relationship with the person whose information is on the D-Card. Thus, a D-Card has an entirely different purpose than that of a vCard. With this dedicated information, a user's relationship with his friends can be verified and leveraged to establish trusted communications with other users.

As shown in Figure 2, the D-Card provides two types of information: (1) the user's general contact information; and (2) the dedicated information that describes the fixed relationship between the user and his friend.

In part (1), a user uniformly defines his contact information for his friend in a manner similar to a vCard's definition thereof. The format is:

Name || Phone Number

Users can extend this format to include more contact information.

In part (2), a user uniquely defines his relationship with each of his friends, creates a public key to verify his digital signature for each of them, generates a digital signature based on the relationship information and each friend's Bluetooth address, and sets each signature's expiration date. The format of this part is:

Relationship || Public Key || Bluetooth Address ||
(Digital Signature) Expiration Date || Digital Signature

One benefit of the dedicated information is the relationship definition, which helps to meet the design goal of privacy. Based on this information, users can control the level of information exchanged between different friends. Another benefit is the digital signature. Due to the absence of a central certificate authority in the Bluetooth piconet, our proposed digital signature can play the role as a local trusted authority to make a user's Bluetooth address and his relationship with other users verifiable.

The dedicated information is not limited to the friend relationship information and Bluetooth address. A user can put any information here, and his digital signature makes it verifiable. For example, the information can be a description of a user's action or physical character. Furthermore, mobile phones' pocketable form factor facilitates carrying and using many D-Cards.

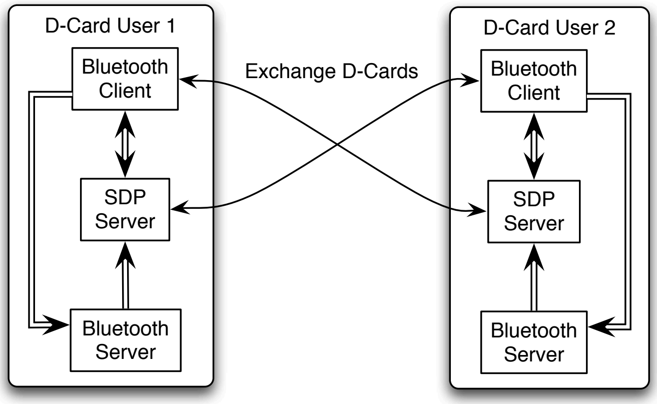


Fig. 3. D-Card Context Exchange.

D. D-Card Exchange

In this subsection, we discuss the D-Card exchange process.

A straightforward method to verify friendships is direct comparison of each user's contacts based on name, phone number, etc. However, this method has two problems. First, there is no privacy protection for the exchanged contact list, so any third party can eavesdrop on the communication and directly obtain sensitive information. Second, storing all contact information entails a large message size. This is inefficient, especially in transmission.

Alternatively, we could use a Bluetooth device name to store and exchange contact and dedicated information. Such names can store at most 248 bytes of text data. However, this method is also problematic, as it provides no privacy for the exchanged data. Also, Bluetooth device names afford us far less flexibility than Bluetooth SDP. Therefore we use SDP to store and exchange this information.

We leverage our Bluetooth SDP toolkit [3] and techniques developed in [22] to exchange D-Cards via Bluetooth without establishing a connection. This enables strangers encountering each other to exchange D-Cards and verify friendships without building connections, which is tedious and inappropriate for these encounters. The D-Card exchange is depicted in Figure 3. We start two virtual services, one for the D-Card contact information and the other for the D-Card dedicated relationship information.

To provide user privacy and power efficiency, we use a Bloom filter [2] to store the contact information. The Bloom filter is a time- and space-efficient probabilistic data structure that tests whether an element is a member of a set. Specifically, the filter is a bit vector of length m in which n inserted items are hashed using k independent hash functions. Bloom filters provide fast lookups and insertions and offer a very compact representation of the set of values being stored. They use one-way hashing to store the elements, which makes it very difficult

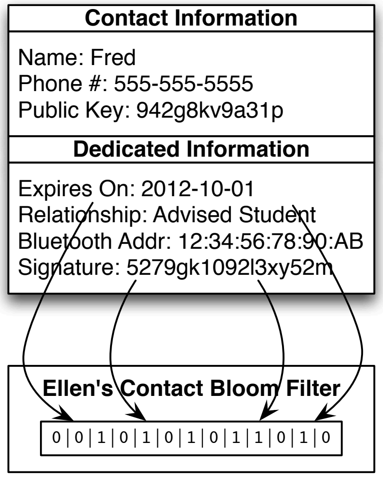


Fig. 4. Bloom Filter Encoding.

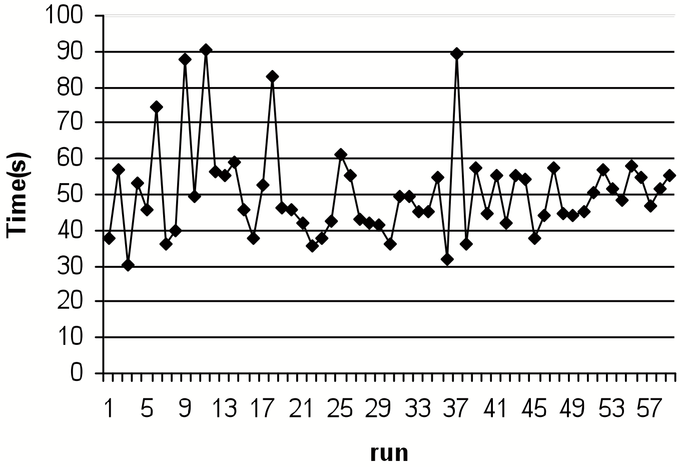


Fig. 5. Friend Discovery Time.

to reconstruct the list of elements in a filter without exhaustively searching the set of all possible elements.

We generate our input string for the Bloom filter by concatenating each field of the first part of the D-Card. After applying several hash functions to the input string, we treat the result as an offset into the bit vector and set the corresponding position to "1". If the bit is already set, we leave it as "1". Suppose Fred is a student who Ellen advises. Then his contact information (name and phone number) can be encoded into Ellen's Bloom filter as shown in Figure 4.

While Bloom filters are time- and space-efficient, there are false positives in their lookup operations. Since the hash functions used by a Bloom filter are not collision-resistant, it is possible that two values are mapped to the same k positions in the bit vector. Specifically, if n items are to be inserted and hash functions select filter positions with equal probability, the false positive rate is

$$f = (1 - (1 - 1/m)^{kn})^k \approx (1 - e^{-kn/m})^k.$$

We select the value of k as follows. The maximum SDP attribute length in one of our phones is $m = 125$ bytes. We conduct a survey of college students and determine that they have an average of $n \approx 150$ contacts in their mobile phones. As we aim to achieve a false positive rate of $f = 1\%$, we solve the equation and choose $k = 7$ independent hash functions.

E. Implementation

We choose Java ME as our prototype development environment because the many mobile phones on the market support Java ME programs with a Java virtual machine. We implement the D-Card CPNS with the Eclipse SDK and the Sun Java Wireless Toolkit for Connected Limited Device Configuration (CLDC) based on the Mobile Information Device Profile (MIDP) specification. We use the Java APIs for Bluetooth described in the JSR-82 interface for developing service discovery applications. We import some Java code from the XSiena (eXtended Scalable Internet Event Notification Architecture) project [21] for the hash functions used in the Bloom filter. The size of the MIDlet application JAR file is about 127KB. We deploy the system on several brands of smartphones, including Sony Ericsson (W810i) and Nokia (5610xm, 6650, N70, N75, N82).

As we mentioned before, D-Card consists of two parts: users' general contact information and dedicated information. The former part stores personal information such as one's name, phone number, etc. In the latter part, we use the relationship between the D-Card owner and holder, the D-Card's expiration date, and the D-Card holder's Bluetooth MAC address to determine the D-Card's validity. To protect this dedicated information, we use the Elliptic Curve Digital Signature Algorithm (ECDSA) to generate and verify the digital signature from the D-Card information. By using ECDSA, the D-Card system can achieve the same security as RSA with significant storage space savings for the digital signatures. When we implemented ECDSA, we set parameters from the `lcrypto` test package that corresponded to a 191-bit binary elliptic curve model. Based on these features, the generated D-Card becomes a unique name card issued from one friend to another designated friend, and this pair of friends will acquire and store each other's D-Cards when their friendship is established.

To minimize the number of required hash functions, we use the technique described in [8]. By doing so, we only need to use 2 hash functions to achieve the same performance of $k = 7$ hash functions.

IV. Evaluation

We evaluate our prototype D-Card CPNS using two measurement metrics, i.e., the *friend discovery time* and the *power consumption rate*. Friend discovery time is defined as the period that from the time of starting a search to the time of finding a potential friend who is within the user's vicinity (Bluetooth communication range). This can be considered an end-to-end delay. Power consumption is measured when we continuously run our application and examine the percentage of power consumed during a given period of time.

We use a Sony Ericsson W810i mobile phone and a Nokia 5610XM mobile phone in our experiments. We run our applica-

Time (min)	0	30	60	90	120
Power (%)	38	36	33	29	26
Power Consumption (%)	–	2	3	4	3

TABLE I
POWER CONSUMPTION.

tion for two hours. A search is started every two minutes. Figure 5 shows the friend discovery time versus the run times of our experiment. The x-axis is the number of runs and the y-axis is the discovery time of the corresponding run. The maximum discovery time is 90.29 seconds, the minimum discovery time is 30.53 seconds, and the average time is 50.52 seconds.

There are several components of the friend discovery time, which is shown in Figure 5. The first component is the time used to inquire for nearby Bluetooth-enabled mobile phones. According to the Bluetooth protocol, this step takes at most 10.24 seconds. A Bluetooth device will not respond to another such device when it is inquiring for nearby devices. This means if two mobile phones start searching for devices at approximately the same time, they will not obtain each other's service information later. The second and third components are the two times of service discovery. Two mobile phones need to make two Bluetooth SDP requests in order to fetch (encoded) D-Cards. These two service discoveries occur rather quickly. However, in order to avoid collisions in the first step and have enough time for other phones to determine nearby D-Card users and publish the dedicated information, we adopt a random (15–40 second) delay between these two service search procedures. Other components are processing and verifying the digital signature using the ECDSA.

Table I shows the power status when our D-Card application is running. The initial power status is 38% when we start the application. After running it for two hours, it is 26% when we stop the experiments. We also keep the mobile phone active by pressing any key whenever it enters power saving mode. The power status is recorded every 30 minutes. The percentage of power consumption is shown in the second row. On average, our D-Card application consumes 6% of the phone's power per hour when we start a search every two minutes. These results are encouraging as users will only run D-Card when necessary, e.g., in particular social settings.

Future Work: For user experience reasons, it is important to reduce the D-Card friendship verification time while performing such verification among strangers in an unobtrusive way. We point to techniques proposed by Woodings et al. [20] and Scott et al. [5] that reduce Bluetooth device discovery time; these techniques can be leveraged to improve system performance. We can also leverage techniques such as Point&Connect [15] to indicate nearby people with whom friendship verification should be performed, which obviates device discovery. Unobtrusively indicating such people and rapidly verifying friendships with them form important aspects of our future work.

V. CONCLUSION

This paper discussed social CPNSs that bridge physically embodied people with cyber social networking services. Such CPNSs also run on mobile phones. We have presented D-Card, a purely distributed mobile phone based social CPNS that provides friendship verification. D-Card encoded relationship information in an electronic name card as well as a public key and digital signature for such verification. We leveraged a Bluetooth SDP toolkit to exchange such name cards in a connectionless manner. We implemented our D-Card CPNS in Java ME and Bluetooth with which many mobile phones are equipped. Our experiments on real-world mobile phones illustrated its promise in realizing distributed mobile phone based friendship verification.

REFERENCES

[1] T. Ahonen. 5-4-3-2-1, as in Billions. What do these gigantic numbers mean?, 6 Aug. 2010. http://communities-dominate.blogs.com/brands/2010/08/5-4-3-2-1-as-in-billions-what-do-these-gigantic-numbers-mean.html.

[2] B. H. Bloom. Space/Time Trade-Offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13:422–426, July 1970.

[3] Bluetooth SDP Toolkit. http://www.cse.ohio-state.edu/~tengj/btsdp.

[4] A. Burak and T. Sharon. Usage Patterns in FriendZone – Mobile Location-Based Community Services. In *Proc. 3rd Int'l. Conf. on Mobile and Ubiquitous Multimedia*, Oct. 2004.

[5] D. Scott and R. Sharp and A. Madhavapeddy and E. Upton. Using Visual Tags to Bypass Bluetooth Device Discovery. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(1):41–53, 2005.

[6] Facebook. http://www.facebook.com.

[7] L. E. Holmquist, J. Falk, and J. Wigström. Supporting Group Collaboration with Interpersonal Awareness Devices. *Personal Technologies*, 3:13–21, 1999.

[8] A. Kirsch and M. Mitzenmacher. Less Hashing, Same Performance: Building a Better Bloom Filter. In Azar, Yossi and Erlebach, Thomas, editor, *Proc. European Symp. on Algorithms (ESA)*, volume 4168 of *Lecture Notes in Computer Science*, pages 456–467. Springer Berlin/Heidelberg, 2006.

[9] J. Kjeldskov and J. Paay. Just-for-Us: A Context-Aware Mobile Information System Facilitating Sociality. In *Proc. 7th Int'l. Conf. on Human Computer Interaction with Mobile Devices & Services*, Sept. 2005.

[10] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold. PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones. In *Proc. 6th Int'l. Conf. on Mobile Systems (MobiSys)*, June 2008.

[11] LinkedIn. http://www.linkedin.com.

[12] MobiLuck. http://www.mobiluck.com.

[13] MySpace Inc. MySpace. http://www.myspace.com.

[14] A. Oulasvirta, M. Raento, and S. Tiitta. ContextContacts: Re-Designing SmartPhone's Contact Book to Support Mobile Awareness and Collaboration. In *Proc. 7th Int'l. Conf. on Human Computer Interaction with Mobile Devices & Services*, Sept. 2005.

[15] C. Peng, G. Shen, Y. Zhang, and S. Lu. Point&Connect: Intention-based Device Pairing for Mobile Phone Users. In *Proc. Int'l. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, pages 137–150, 2009.

[16] R. Richmond. Stolen Facebook Accounts for Sale, *The New York Times*, http://www.nytimes.com/2010/05/03/technology/internet/03facebook.html, 2 May 2010.

[17] Sophos. Facebook, Fake AV, and Friends, http://nakedsecurity.sophos.com/2008/11/26/facebook-fake-av-and-friends/, 26 Nov. 2008.

[18] vCard. http://en.wikipedia.org/vCard.

[19] Wireless Intelligence. Global mobile subscriptions surpass 5 billion milestone, 8 July 2010. https://www.wirelessintelligence.com/print/snapshot/100708.pdf.

[20] R. Woodings, D. Joos, T. Clifton, and C. D. Knutson. Rapid Heterogeneous Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pages 342–349, 2002.

[21] xSiena. http://wwwse.inf.tu-dresden.de/xsiena/.

[22] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li. E-SmallTalker: A Distributed Mobile System for Social Networking in Physical Proximity. In *Proc. of IEEE Int'l. Conf. on Distributed Computing Systems (ICDCS)*, 2010.