BEDAH PAPER: SOLUTIONS TO SECURITY AND PRIVACY ISSUES IN MOBILE SOCIAL NETWORKING *

Yoenie Indrasary [(1)] dan Siti Kustini [(1)]

[(1)] Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Banjarmasin

Ringkasan

Pada paper ini akan dipresentasikan permasalahan privasi dan keamanan yang signifikan yang muncul pada hampir seluruh sistem jejaring sosial bergerak, khususnya yang berbasis konteks lokasi (Location-Aware Mobile Social Network-LAMSN). Paper ini membuat tiga masukan utama dalam makalah ini yakni: 1. Mengidentifikasi bahwa terdapat tiga macam masalah privasi dan keamanan pada sistem jejaring sosial bergerak: (1) persoalan direct anonymity, (2) per-soalan indirect atau K-anonymity, dan (3) serangan eavesdropping, spoofing, replay, dan wormhole attacks. Di sini akan dibahas bagaimana persoalan tersebut dapat memunculkan tantangan tersendiri untuk kasus sistem jejaring sosial bergerak. 2. Mengetengahkan rancang-an untuk suatu sistem, yang dinamakan server identitas (identity server), yang menyediakan solusi bagi masalah keamanan dan privasi ini. 3. Menjelaskan implementasi server identitas yang dimaksud.

Kata Kunci *: Security, privacy, Location-Aware Mobile Social Network (LAMSN), anonymity, Identity Server*

1. PENDAHULUAN

Fokus adalah pada keamanan dan privasi dalam sistem *location-aware mobile social network* (LAMSN). Jejaring sosial online saat ini memiliki jutaan pengguna. Hubungan sosial dan informasi seputar preferensi user-nya memungkinkan dikembangkannya aplikasi turunan dari situs jejaring sosial tersebut. Lebih jauh lagi, informasi jejaring sosial saat ini telah dikorelasikan dengan lokasi fisik user, yang memungkinkan preferensi user dan aspek sosial-nya berinteraksi secara real-time dengan lingkungan fisiknya.

Sistem LAMSN seperti WhozThat [1] dan Serendipity [2] menyediakan infrastruktur untuk meningkatkan konteks jejaring sosial dalam kedekatan fisik lokal menggunakan *mobile smartphones*.

Bagaimanapun, sistem tersebut biasanya hanya memberikan sedikit perhatian terhadap keamanan dan privasi dari informasi personal para user dari situs jejaring sosial tersebut.

2. LATAR BELAKANG DAN UPAYA SEBELUMNYA YANG BERKAITAN

Pada bagian ini diberikan pendahuluan singkat mengenai hal yang terkait dengan jaringan sosial bergerak (mobile social networking) dan teknologi yang membuatnya terimplementasi.

*) Aaron Beach, Mike Gartrell, dan Richard Han
University of Colorado at Boulder
International Conference on Computational Science and Engineering

Mobile Computing

Smartphones saat ini memungkinkan jutaan orang terhubung ke internet setiap saat, dan memberikan dukungan lingkungan pengembangan yang mapan bagi *developer* aplikasi (pihak ke-3).

Dukungan teknologi yang tersedia pada *smartphone* mampu mengalihkan proses perhitungan ke saku user yang juga berarti memberikan *ubiquitous access* untuk informasi jejaring sosial online bagi user.

Aplikasi Jejaring Sosial Bergerak *(Mobile Social Network)* yang Telah Ada

Tantangan unik jejaring sosial bergerak yang dijelaskan di sini banyak diperoleh dari hasil pekerjaan penulis pada aplikasi *WhozThat* [1] dan SocialAware [3].

Keduanya adalah sistem terdahulu yang memungkinkan pembuatan aplikasi *context-aware* (*location-aware*) yang mengeksploitasi informasi jejaring sosial dari jejaring sosial online yang telah ada seperti Facebook.

Telah banyak aplikasi yang menggunakan pendekatan tradisional dan sederhana dalam mengintegrasikan informasi jejaring sosial dengan lokasi user dan informasi konteks. Bentuk aplikasi yang paling umum adalah yang secara sederhana memperluas akses jejaring sosial dari desktop ke telpon bergerak, atau menyediakan antarmuka jejaring sosial yang telah dioptimasi untuk dapat diakses dari telepon bergerak/berpindah. Contoh dari aplikasi Sosial Network tersebut adalah iPhone atau Blackberry Facebook.

Ada juga yang menggunakan pendekatan sensor jaringan yang mendayagunakan fitur – fitur khusus pada *mobile phone* untuk menjadi informasi konteks lokal bagi jejaring sosial. Sebagai contoh CenceMe yang mengirim informasi konteks ke jejaring sosial (mis. lokasi user, dan mungkin *context cues* untuk mengetahui apakah user sedang berbicara atau tidak [4]). Pendekatan ini agak *unidirectional*, yakni berfokus pada memperkaya jejaring sosial (dan aplikasi desktop-nya) melalui konteks user.

Sebaliknya, sistem yang dirancang untuk WhozThat mengeksploitasi teknologi komputasi bergerak (*mobile computing*) untuk meng-import informasi kontekstual dari situs jejaring sosial ke lingkungan fisik lokal user. Serendipity [2] adalah sistem yang mirip dengan WhozThat, ia meng-import konteks sosial ke konteks lokal menggunakan perangkat bergerak. Namun Serendi-pity mengelola database tersendiri untuk informasi konteks sosial-nya dan tidak berkoneksi dengan situs jejaring sosial online populer yang lain.

3. MASALAH KEAMANAN DAN PRIVASI

Sistem jejaring sosial bergerak dengan konsep peer-to-peer seperti WhozThat dan Social Aware, mempertukarkan pengenal identitas user jejaring sosial di antara perangkat bergerak menggunakan teknologi *short-range wireless* seperti Bluetooth. Sementara sistem jejaring sosial bergerak semacam Brightkite dan Loopt, akan memberitahu server tersentralisasi (*centralized server*) mengenai lokasi terakhir dari perangkat (tersedia melalui GPS, identifikasi *cell-tower*, atau mekanisme lain). Dengan meng-*query* server, perangkat bergerak dalam sistem *client-server* ini dapat menemukan user terdekat beserta informasi lain yang berkaitan.

Berikutnya akan dibahas masalah privasi dan keamanan yang berkaitan dengan sistem jejaring sosial bergerak atau berpindah yang menggunakan model *peer-to-peer* dan *client-server*.

Persoalan *Direct Anonymity*

Model pertukaran informasi pada sistem jejaring sosial bergerak yang didiskusikan sebelumnya memberikan perlindungan yang rendah terhadap privasi user. Sistem ini mensyaratkan user untuk mengijinkan akses terhadap informasi profil-nya yang ada pada jejaring sosial dan pada saat yang sama menghubungkan informasi tersebut terhadap identitas user.

Sebagai contoh, aplikasi Facebook biasanya meminta user untuk memberikan akses terhadap informasi mereka (user) melalui API Facebook, yang kemudian secara intrinsik mengaitkan informasi ini ke identitas user. Pada sistem WhozThat dan SocialAware, setiap orang yang berada dekat user bergerak (*mobile user*) dapat menggunakan perangkat *Bluetooth* untuk menyadap ID jejaring sosial user atau melakukan *eavesdrop* pada data yang dikirimkan secara terbuka melalui koneksi nirkabel, meskipun provisi enkripsi yang lemah pada layer link tersedia dalam media komunikasi yang digunakan ini[].

Dalam sistem jejaring sosial bergerak context-aware yang bersifat peer-to-peer, kita dapat melacak user dengan melakukan proses logging terhadap tanggal dan waktu ketika setiap perangkat bergerak atau stasioner mendeteksi ID user jejaring sosial. Dengan mengumpulkan log tersebut, kita dapat memiliki *history* dari lokasi yang pernah dikunjungi user, dan waktu pada tiap kunjungan.

Pada akhirnya, dengan adanya akses terhadap ID jejaring sosial user, orang lain dapat mengakses informasi publik user tersebut tanpa sekeinginannya. Kami menyimpulkan bahwa pertukaran *cleartext* dari ID jejaring sosial seperti WhozThat dan SocialAware mengarah pada resiko keamanan dan privasi yang tidak dapat diterima, dan memungkinkan *anonymity* user. Kami menyebut permasalahan yang secara langsung memungkinkan *anonymity* user sebagai serangan *direct anonymity*.

Direct anonymity attacks juga dapat muncul pada sistem jejaring sosial bergerak yang menggunakan teknik *client-server*. Jika biasanya pada sistem client-server ID user jejaring sosial tidak secara langsung dipertukarkan, perangkat bergerak atau stasioner masih dapat melacak user dengan melakukan logging terhadap tanggal dan waktu ketika setiap perangkat menemukan user di sekitarnya. Karena tiap perangkat pada sistem ini dapat menemukan nama user jejaring sosial yang biasanya berupa nama lengkap, privasi user dapat dikompromikan.

Dari sini kita memperoleh isu *direct anonymity* ; yakni ekspos lokasi dan nama user pada sistem *client-server* memungkinkan *anonymity* user dikompromikan.

Masalah *Indirect* atau *K-Anonymity*

Tantangan lainnya adalah bagaimana mendukung aplikasi jejaring sosial bergerak yang melibatkan informasi personal tanpa mengkompromikan *anonymity* dari user yang memberikan informasi tersebut. Bahkan jika user tidak secara langsung menyediakan informasi identifikasinya, informasi user jejaring sosial yang telah tersedia (misalkan preferensi) dapat dipetakan kembali terhadap identitas user melalui situs jejaring sosial atau informasi yang di-*cache* dalam perangkat bergerak/stasioner dalam lingkungan. Masalah *Indirect atau K-anonymity* muncul

ketika potongan informasi atau set informasi yang berkaitan dapat digunakan bersama-sama untuk secara unik memetakan kembali identitas user. Contoh dari ini adalah ketika beberapa potongan informasi unik dari user diberikan, seperti daftar film favorit dan semacamnya, kemudian dapat dengan mudah dipetakan kembali ke user tertentu.

Lebih jauh lagi, jika set informasi hanya dapat dipetakan ke set ***k*** atau set user yang lebih kecil, *anonymity* user masih dapat dikompromikan pada derajat yang bersesuaian dengan ***k***.

Tantangannya adalah bagaimana merancang algoritma yang dapat memutuskan informasi macam apa yang perlu atau tidak perlu diberikan, dalam rangka menjamin *anonymity* dari user yang bersangkutan.

Beragam dan banyaknya informasi di jejaring sosial membuat jaminan privasi user lebih rumit daripada yang mungkin terlihat. Secara lebih formal, permasalah spesifiknya adalah menemukan informasi personal macam apa yang dapat dibagi sedemikian sehingga informasi ini tidak dapat digunakan untuk mengaitkan identitas user dengan konteks tertentu.

Pada beberapa kasus *K-anonimity* sebelumnya telah ditunjukkan bahwa dengan mengkorelasikan beberapa set data, sejumlah besar record dapat di-identifikasi kembali. Sebuah paper oleh *Sweeney* menunjukkan bagaimana proses re-identifikasi dilakukan menggunakan record voter dan record rumahsakit [5].

Masalah *K-anonymity* dalam paper ini bersifat unik dalam pengertian bahwa standar *K-anonymity* menjamin informasi yang diberikan tidak dapat membedakan ke- ***k - 1*** individu yang terkait dengan informasi yang dilepaskan. Namun, masalah yang didiskusikan dalam paper ini tidak melibatkan pelepasan record personal melainkan lebih berupa se-set informasi yang ter-agregasi yang dapat berkaitan dengan seset individu yang dapat atau tidak dapat berhubungan dengan informasi yang dilepaskan.

Karenanya, jaminan *K-anonymity* pada permasalahan kita mengacu pada jumlah "minimal" set unik *indistinguishable* yang masih mencukupi untuk seluruh informasi yang dilepaskan. Untuk lebih tepatnya, tidak boleh ada lebih dari ***k-1*** set unik yang bukan merupakan subset dari satu sama lain, dan seluruh set lain adalah superset dari sebagian set minimal.

Set "minimal" dari set ekivalen didefinisikan dengan penyederhanaan dari ekspresi aljabar Boolean, di mana elemen dari seluruh set yang cukup, terhubung dengan penghubung (AND) dan seluruh set secara logika *disjunct* (OR). Bentuk penyederhanaan dari ekspresi ini didefinisikan sebagai set "minimal" dari set di mana ekspresi yang disederhanakan terdiri atas lebih dari ***k-1*** set yang terpisahkan secara logika. Sekumpulan data dengan ***k-1*** set minimal dapat diterima di bawah jaminan *K-anonymity* dari k.

Masalah ini dapat dinyatakan sebagai masalah *admissible set*.

Diberikan dua set A dan B dengan A adalah set dari seluruh user dan B adalah set dari seluruh informasi jejaring sosial yang dapat diberikan bagi aplikasi jejaring sosial bergerak. Informasi dalam B memiliki relasi banyak-ke-banyak terhadap A, karena user dapat memiliki banyak informasi yang terkait dengannya dan banyak user dapat berhubungan dengan potongan informasi yang identik.

Masalahnya kemudian adalah mendefinisikan *admissible set* di bawah jaminan *K-anonymity* , yang akan mendefinisikan apakah subset ***x*** dari B *admissible*.

Paper ini mengetengahkan masalah *K-anonymity* secara informal dan mengajukan solusi yang saat paper ini dibuat tengah diteliti dan diimplementasikan oleh penulisnya. Kami memandang isu ini penting karena ia akan memberikan alternatif bagi user untuk memanfaatkan aplikasi jejaring sosial bergerak yang baru tanpa merugikan privasi mereka.

Masalah *K-anonymity* berlaku baik pada sistem peer-to-peer maupun *client-server*, karena kedua sistem melibatkan sharing profil user jejaring sosial yang satu terhadap user jejaring sosial yang lain.

Eavesdropping, Spoofing, Replay, dan *Wormhole Attacks*

Begitu ID user jejaring sosial telah di-*intercept* pada sistem peer-to-peer sistem jejaring sosial bergerak, ia dapat digunakan untuk me*mount* serangan *spoofing* dan *replay*. Dalam serangan *spoofing*, user tak dikenal dapat menyamar sebagai user yang ID-nya telah di-*intercept* (user yang telah mengkompromikan privasi-nya) dengan hanya mengirim (*replaying*) ID yang telah di- *intercept* ke perangkat bergerak atau stasioner yang meminta ID user jejaring sosial. Maka, serangan *replay*-nya, di mana ID user secara liar diulang, digunakan untuk melakukan *spoofing*. Jenis spesifik lain dari serangan *replay* dikenal sebagai serangan *wormhole* [10], di mana transmisi nirkabel di-*capture* pada salah satu ujung dari jaringan dan di-*replay* pada ujung yang lain. Pada sistem semacam Whoz-That atau SocialAware, user tak-resmi dapat menggunakan serangan *wormhole* untuk menangkap ID user dan menyamar sebagai user dengan lokasi, atau mungkin jarak, yang berbeda. Karena sistem ini rentan terhadap serangan *replay* dan *spoofing,* kita tidak dapat lagi percaya bahwa tiap user yang berpartisipasi dalam sistem ini adalah benar-benar sesuai dengan yang mereka klaim. Karenanya, nilai dari sistem tersebut secara substansial telah hilang.

Lebih jauh lagi, serangan ini dapat digunakan untuk tujuan lain. Sebagai contoh, user tak-resmi dapat menyamar sebagai user lain pada tempat dan waktu tertentu ketika ia melakukan kejahatan.

Sudah jelas bahwa serangan *spoofing* pada sistem jejaring sosial bergerak memberikan resiko keamanan yang serius. Tidak hanya meng-*intercept* ID user jejaring sosial melalui *eavesdropping* pada jaringan nirkabel, user tak-resmi dapat melakukan *eavesdrop* pada informasi yang ditransmisikan ketika perangkat meminta informasi profil user jejaring sosial dari server jejaring sosial.

Sebagai contoh, jika perangkat bergerak dalam sistem peer-to-peer menggunakan HTTP (RFC 2616) untuk berhubungan ke server API REST Facebook dan bukannya ke HTTPS (RFC 2818), seluruh informasi user yang diminta melalui server API Facebook ditransmisikan dalam *cleartext* dan dapat di-*intercept.* Intercept dari data memungkinkan user tak-resmi mengakses informasi pribadi user yang sebenarnya tidak dimaksudkan untuk di-*share*. Namun serangan *eavesdropping, spoofing, replay*, dan *wormhole* secara umum bukan ancaman utama pada sistem jejaring sosial bergerak. Serangan ini dapat dilawan dengan penggunaan yang tepat atas protokol keamanan yang *robust* seperti HTTPS.

Jika *credential* dari login jejaring sosial user belum dicuri oleh user tak-resmi, maka hampir mustahil user tak-resmi tersebut mampu menyamar sebagai user yang asli dalam jejaring sosial tersebut..

4. SOLUSI KEAMANAN DAN PRIVASI

Di sini penulis paper terkait telah merancang dan mengimplementasikan sistem yang dinamakan *identity server*, untuk menyikapi masalah privasi dan keamanan sebagaimana dijelaskan sebelumnya. Sistem ini mengasumsikan setiap perangkat bergerak yang berpartisipasi memiliki akses internet yang andal melalui koneksi data wireless wide area network (WWAN) atau melalui koneksi WiFi.

Kemudian diasumsikan bahwa setiap perangkat bergerak yang berpartisipasi memiliki antarmuka jaringan nirkabel jarak-pendek, seperti *Bluetooth* atau *WiFi*, untuk komunikasi adhoc dengan perangkat bergerak/stasioner sekitarnya.

Rancangan Server Identitas dan *Anonymous Identifier*

Untuk menanggulangi resiko pertukaran ID user jejaring sosial, kami mengajukan penggunaan *anonymous identifier*, atau AID. AID adalah *nonce* yang dibangkitkan oleh server yang terpercaya, yang bernama *identity server* (IS). Sebelum user perangkat bergerak mengumumkan kehadirannya ke user perangkat bergerak/stasioner yang lain, ia akan menghubungi IS untuk memperoleh AID. IS kemudian akan membangkitkan AID baru untuk perangkat bergerak ini menggunakan fungsi hash *cryptographic* seperti SHA-1, dengan nilai pembangkit acak. IS kemudian akan menghubungkan AID yang baru dibuat terhadap perangkat bergerak yang meminta AID tersebut. Kemudian user pemilik AID ini akan melanjutkan men-*share* AID ini terhadap perangkat bergerak/stasioner yang ada di sekitarnya.

Setelah perangkat bergerak atau stasioner yang ada di sekitarnya (perangkat B) mengenali AID ini (perangkat A), ia akan membangun koneksi terhadap A untuk memperoleh AID bersama. Setelah AID telah diperoleh oleh perangkat B, perangkat A akan meminta AID lain pada IS. AID yang baru ini akan di-*share* dengan perangkat bergerak atau stasioner berikutnya yang kemudian akan berkoneksi dengan layanan *sharing* AID pada perangkat A.

Setelah perangkat B memperoleh AID bersama dari perangkat A, perangkat B akan melanjutkan dengan meng-*query* informasi profil jejaring sosial pada IS. Gambar 1 menunjukkan peran IS dalam membangkitkan AID dan memroses permintaan akan informasi user jejaring sosial.

Begitu informasi jejaring sosial telah diberikan oleh IS, IS menghapus AID ini dari daftar hubungan antara AID dan perangkat bergerak user. Sebelum perangkat bergerak user mengumumkan kehadirannya menggunakan layanan bersama AID *Bluetooth,* ia akan memperoleh AID baru dari IS seperti dijelaskan di atas.

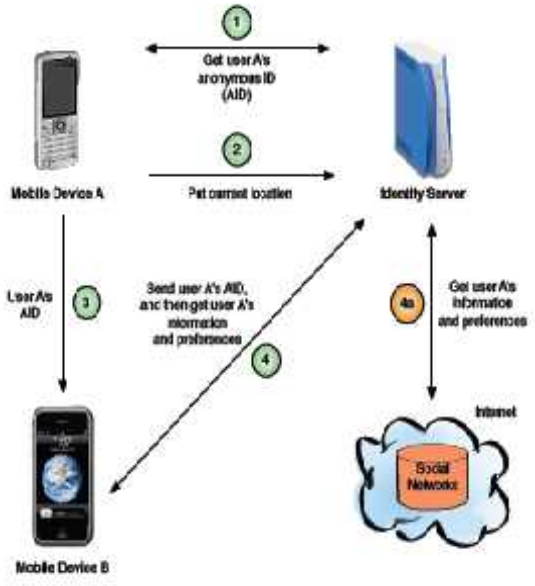
Di sini beberapa AID dapat berkaitan dengan satu user mobile, sehingga memungkinkan beberapa perangkat bergerak/stasioner memperoleh informasi mengenai user. Untuk memperbaiki efisiensi, perangkat bergerak user dapat mengirimkan permintaan lebih dari satu AID ke IS, dan akan terus berbagi AID terhadap perangkat lain yang ada di sekitarnya.

IS akan menge-set nilai timeout untuk tiap AID ketika AID dibuat. AID mencapai *timeout* ketika ia tidak dikonsumsi selama periode *timeout*-nya, yakni, jika IS tidak menerima query atas informasi profil jejaring sosial selama periode *timeout* tersebut.

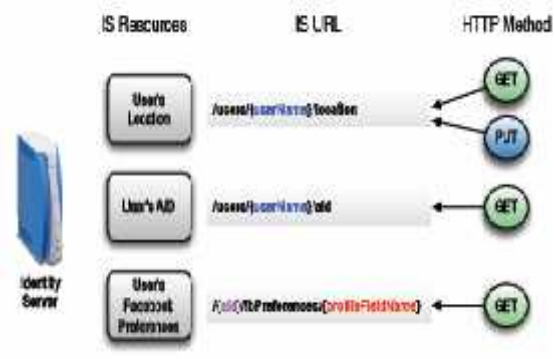
Karena IS tidak mendukung pencarian informasi personal, perangkat yang mencari informasi jejaring sosial untuk user yang terkait dengan AID tidak dapat menghubungkan AID terhadap identitas jejaring sosial user. Jadi hanya dengan mengkompromikan IS user tak-resmi dapat mengkaitkan AID dengan ID jejaring sosial user.

Implementasi *Identity Server*

Dalam paper ini mengimplementasikan IS menggunakan platform Java Standard Edition (SE) 5.0. Seluruh servis IS yang diakses perangkat bergerak maupun stasioner diperlakukan sebagai *web services* menurut arsitektur REST. Kemudian juga menggunakan framework open source Reslet untuk Java untuk mengembangkan IS.



Gambar 1. Ilustrasi mekanisme pembangunan koneksi antar user melalui Identity Server



Gambar 2. Mekanisme penyediaan sumber daya pada Identity Server

Seluruh sumber daya IS seperti AID *mobile user*, lokasi terakhir user, dan informasi profil Facebook untuk *mobile user*, di-ekspos sebagai sumberdaya yang dapat diakses melalui URL dan mendukung *method* HTTP GET, POST, dan PUT. Gambar 2 menunjukkan sumberdaya yang dapat diakses melalui web dan di-ekspos melalui IS.

Body tiap permintaan HTTP di-kode menggunakan JSON (RFC 4627). Seluruh *traffic* jaringan *web service* antara IS dan perangkat bergerak/stasioner lain di-enkripsi menggunakan HTTPS, dan akses terhadap seluruh sumber daya di-autentikasi menggunakan autentikasi akses dasar HTTP (RFC 2617).

Tiap user harus mendaftar ke IS sebelum berpartisipasi dalam sistem jejaring sosial penulis. Selama proses pendaftaran, user memberikan Facebook user ID-nya, dan memilih *user name* dan *password*. *User name* dan *password* tersimpan dengan aman pada perangkat bergerak user, dan digunakan untuk mengautentikasi IS dan memperoleh akses terhadap sumberdaya web yang ada pada IS.

Penulis mengimplementasikan seluruh persistensi data pada IS menggunakan *tool open-source* SimpleJPA. SimpleJPA adalah implementasi *Java Persistence API* (JPA) untuk *Amazon's SimpleDB*.

Dengan menggunakan SimpleDB, dapat diperoleh sistem database terdistribusi yang sederhana, *scalable*, dan andal.

AID untuk tiap *mobile user* dibangkitkan pada IS dengan fungsi hash kriptografik dengan *salt value* acak 16-byte acak.

Facebook REST API *web service* digunakan oleh IS untuk memperoleh isi dari profil Facebook user. Setiap perangkat bergerak/stasioner (B) meminta preferensi user Facebook dari *mobile user* (A), IS memeriksa lokasi perangkat A dan B untuk memverifikasi bahwa perangkat ini berada dalam jarak yang memadai bagi satu sama lain

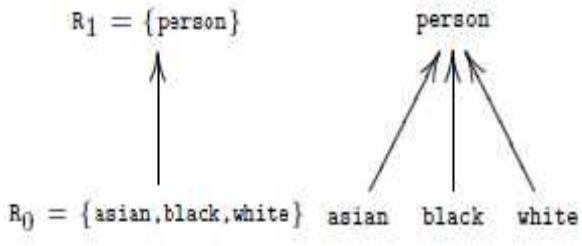
K-Anonymity

Dalam paper ini penulis membuat sebuah analogi untuk menjelaskan masalah *K-anonymity*. Perhatikan contoh data set dalam Tabel I. Jika set (Red; A; 1) dilepaskan atau diberikan pada aplikasi pihak ketiga, ia akan dapat direlasikan kembali ke *minimal unique set* (Bill), dan (Fred; Joe) yang mengimplikasikan bahwa setidaknya, Bill OR Fred AND Joe saling berkaitan dengan data. Ini tidak mengesampingkan kemungkinan superset lain yang mencakup *minimal set* seperti (Bill; Fred) atau (Bill; Fred; Joe), namun, ia menunjukkan bahwa satu atau dua *minimal set* harus berkaitan dengan data. Ini menjadi contoh bagi *K-anonymity* where k <= 2, sedemikian sehingga k-1 minimal set tidak dapat dibedakan.

TABLE I
K-ANONYMITY EXAMPLE DATA SET

Name	Color	Letter	Number
Bill	Red	A	1
Fred	Green	A	2
Jon	Green	B	2
Joe	Red	C	1

Jika hanya ada dua set user yang dipetakan terhadap sepotong data, potongan data yang lain dalam set yang tersedia atau set apapun sesudahnya yang mengandung potongan data yang sama, dapat digunakan untuk membedakan user mana yang terkait dengan data. Karenanya, algoritma untuk menentukan *admissible set* perlu memelihara *state* antara jumlah ***n*** set sekuensial ***x1; x2; :::xn*** dari informasi, ini akan menjamin setidaknya ***k*** *minimal set* dari user selalu tidak dapat dibedakan karena terkait dengan n set sekuensial dari data yang berhubungan ***x1; x2; :::xn***. Penulis juga mengeksplor algoritma simplifikasi logika sebagaimana yang diberikan oleh Quine-McCluskey [9] untuk memecahkan masalah ini dengan lebih cepat. Untuk menggunakan perangkat lunak, simplifikasi logika yang telah ada seperti hubungan antara user dan preferensinya harus dimodelkan sebagai kasus nyata. Salah satunya adalah dengan memodelkan pasangan user dan preferensinya sebagai node pada graf sebagaimana tampak pada Gambar 3.



Gambar 3. Pemodelan untuk memecahkan permasalahan *k-anonimity*

Pada awalnya node tersebut sebagian diatur sesuai keinginan, kemudian tiap node dihubungkan terhadap seluruh node di hadapan maupun di belakang mereka. Set yang merupakan kasus nyata adalah superset seluruh jalur dari awal node sampai akhir node. Setiap jalur memetakan klausa konjungtif dari huruf (satu huruf per node) dalam ekspresi Boolean *disjunctive normal form* (DNF) final. Seluruh jalur/klausa akan menjadi disjungtif yang menyebabkan ekspresi-nya secara umum menjadi benar jika setiap kasus nyata berakhir benar.

Penulis telah mengimplementasikan pendekatan mendasar ini untuk memverifikasi jaminan *K-anonymity* dan memulai uji unjuk kerja menggunakan algoritma Quine-McCluskey [9] untuk penyederhanaan logika. Di sini digunakan aplikasi multimedia context-aware yang mencuplik preferensi media (music dan film) user dalam area lokal. Seluruh *query* user tertuju ke IS, yang mengimplementasikan jaminan *K-anonymity*.

Test awal telah menunjukkan solusi ini sesuai untuk user group sebesar kebanyakan jumlah daftar jaringan pertemanan (terdiri dari 200–300 friend),yang membuat jaminan *K-anonymity* mungkin dengan ***k*** = 20.

Eavesdropping, Spoofing, Replay, dan *Wormhole Attacks*

Solusi keamanan dan privasi yang ditawarkan dalam paper ini memberikan beberapa fitur keamanan yang ditujukan bagi ancaman keamanan sebagaimana yang dipaparkan pada sub-bagian III-C. Penggunaan AID mencegah serangan *spoofing* dan *replay*. Karena AID di-*share* oleh perangkat bergerak (tidak seperti ID jejaring sosial lain seperti Facebook), user tak-resmi tidak dapat men-*spoof* identitas jejaring sosial dari user lain. Melalui penggunaan fungsi hash kriptografik dengan *salt value* acak untuk membangkitkan AID bagi tiap *mobile user* -dan akan terus membangkitkan AID baru begitu AID mengalami *timeout* atau dikonsumsi oleh perangkat lain kita dapat mencegah serangan *replay* di mana user tak-resmi dapat mencoba menangkap nilai AID dan menggunakan kembali sekumpulan nilai AID yang telah di-*share* oleh perangkat bergerak.

Asumsi terhadap sistem posisi yang aman juga mencegah serangan *wormhole* menggunakan AID, di mana user tak-resmi tidak dapat meng-*capture AID* yang di-*share* oleh perangkat bergerak dan mengirimkannya ulang untuk di-*share* dengan perangkat yang berjarak jauh (perangkat B), karena IS akan memverifikasi bahwa perangkat bergerak yang bersesuaian dengan AID yang dimaksud berada dalam jarak yang memadai terhadap perangkat B setiap kali perangkat B berupaya memperoleh informasi user jejaring sosial dengan AID tersebut.

5. PENUTUP

Kesimpulan

Pada makalah ini dicoba diimpelementasikan sebuah entitas bernama Identity Server (IS) yang ditujukan untuk memberikan jaminan keamanan dan privasi pada pengguna sistem jejaring sosial bergerak khususnya dengan konteks lokasi dan informasi tambahan lainnya yang bersifat lokal. Peran AID yang membuat setiap koneksi menjadi makin unik untuk di-*intercept* diharapkan dapat mempersulit *malicious-user* untuk melakukan serangan *spoofing, eavesdrop*, dan *wormhole*. Optimalisasi *mobile computing* dengan proses autentikasi yang berjalan lokal di perangkat pengguna diharapkan juga makin mem-proteksi privasi ID user. Sayangnya pada permasalahan k-anonimity paper ini belum menggambarkan bagaimana pengaturan informasi preferensi user yang diterapkan untuk dapat memberikan jaminan yang lebih baik agar sekumpulan informasi preferensi tersebut tidak dapat dipetakan secara unik ke user tertentu.

6. DAFTAR PUSTAKA

1. Anonim, (tanpa tahun),."*Simple JPA -Java Persistence API for Amazon SimpleDB*" http://code.google.com/p/simplejpa/.[6]
2. Anonim, (tanpa tahun)."*Java Persistence APIi*" http://java.sun.com/javaee/technologies/persistence.jsp. [7]
3. Anonim, (tanpa tahun)..“*Amazon SimpleDB*" http://aws.amazon.com/simpledb/.[8]
4. Anonim, (tanpa tahun). “Quine-*mccluskey Algorithm (Java)*" http://en.literateprograms.org/ Quine-McCluskey algorithm (Java). [9]
5. A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han, (2008), “*Whozthat? Evolving an Ecosystem for Context-Aware Mobile Social Networks*” IEEE Network, vol. 22, no. 4, pp. 50–55, July-August 2008. [1]
6. N. Eagle and A. Pentland, (2005), “*Social serendipity: Mobilizing social software*,” IEEE Pervasive Computing, vol. 4, no. 2, April-June 2005. [2]
7. Charles M. Gartrell, (2000),“*SocialAware: Context-Aware Multimedia Presentation via Mobile Social Networks*”, Master Thesis at B.S., Virginia Tech. [3]
8. Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, dan Andrew T. Campbell, (2007), “*CenceMe – Injecting Sensing Presence into Social Networking Applications*”, Springer-Verlag Berlin Heidelberg. [4]
9. Latanya Sweeney, (2002), *“k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY*”, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, May 2002. [5]
10. Ritesh Maheshwari, Jie Gao and Samir R Das, (tanpa tahun) ”*Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information*”, Department of Computer Science, Stony Brook University. [10]