A Dynamic Game Analysis of Pseudonym Changes for Location Privacy

Zarko Milosevic
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland

Supervised by: Julien Freudiger and Hossein Manshaei
Professor: Jean-Pierre Hubaux
Course: Security and Cooperation in Wireless Networks
Date: January 22, 2009

1 Introduction

Authentication (i.e. the possibility to verify the identity of communicating parties) is a prerequisite to provide security in mobile ad hoc networks (vehicular networks [9], delay tolerant networks[5], etc). This is usually achieved using asymmetric cryptography. Mobile nodes are preloaded with an asymmetric key pair consisting of one public key and one private key. The private key is used to sign messages and the public key is sent along with messages to verify the signatures. An eavesdropping adversary can track the location of the nodes by monitoring public keys in messages.

One possible solution for achieving location privacy is the multiple pseudonym approach. It is first proposed in the context of Internet communications [4] and then it is adopted as a solution for achieving location privacy in mobile ad hoc networks ([1], [2], [3], [10], [13]). With the multiple pseudonym approach, every node has a set of asymmetric key pairs that can be either pre-loaded from an off-line certification authority [14], or directly generated by mobile nodes [11]. Over time, mobile nodes change their key pairs and there is only a single key pair active at the time. The active private key is used to sign messages and the active public key is used by receiving party to verify the signature. The public key is also used as an identifier of the node and is called the pseudonym. The particularity of this approach is that an individual changing pseudonyms alone cannot achieve location privacy: if a node \( i \) change a pseudonym at time \( t \), then the adversary can trivially link a new pseudonym with the old one. To prevent the adversary from the possibility of linking old and new pseudonyms a collective effort from mobile nodes is needed [6]. More precisely, it is necessary that a change of pseudonym is spatially and temporary coordinated among mobile nodes [1] inside regions called mix zones.

There already exists several proposals in the literature to coordinate the pseudonym change. One solution [2] propose to change pseudonyms periodically at pre-determined frequency. This is not an efficient mechanism because it requires that at least two mobile nodes change their pseudonyms periodically at pre-determined frequency. This is not so realistic assumption because the pseudonym change has a cost:

...
• pseudonyms are costly to acquire
• pseudonym change forces routing algorithms to frequently update their routing tables [14].

In the presence of selfish mobile nodes, selfish nodes might thus not want to change their pseudonyms in the settings that offer low location privacy guarantees.

In contrast with these approaches, the authors of [6] assume selfish mobile nodes that locally decide whether to change their pseudonyms or not. They investigate the efficiency of the multiple pseudonym approach in non-cooperative scenarios. To do so, they propose the first user-centric location privacy model that gives the possibility to capture how the location privacy of a node evolves over time. They also define a game-theoretic model called the pseudonym change game to model the decisions of mobile nodes in a mix zone. In [6] they study the static games with complete (every node has a complete information about the location privacy level of other nodes) and incomplete information. In the static games, nodes meet in a mix zone and simultaneously play their moves. In this report, we use the same pseudonym change game model but instead analyze dynamic games in which players play their moves sequentially. With dynamic games, players have the possibility to observe moves of other players before deciding whether to change a pseudonym or not. We analyze dynamic games theoretically to obtain the Nash equilibria [12] for games with complete information and to obtain the Bayesian Nash equilibrium [8] for games with incomplete information. The results obtained by theoretical analysis are validated with Matlab simulations in which we compare the efficiency of static games compared to dynamic games using two metrics: (i) the average location privacy gain and (ii) the average percentage of players that misused pseudonyms. Using the knowledge gained from game theory analysis we propose a new complete changing pseudonym protocol for achieving location privacy that is based on the Swing protocol from [swing paper]. We compare the efficiency of the simplified version of the Swing protocol with the new protocol using simulations in Matlab and show that the new protocol has a better performances with respect to the metrics (i) and (ii).

The rest of the report is structured as follows. The system model and the adversary model are explained in Section 2. Section 3 defines the pseudonym change game. The game analysis is done in Section 4. In Section 5, the algorithm for dynamic ordering of players for dynamic games with incomplete information is introduced. The efficiency of different approaches is studied using simulations in Matlab in Section 6. Section 7 describes the new changing pseudonym protocol for achieving location privacy called Non-cooperative Swing protocol. Section 8 concludes the paper.

2 System model and adversary model

We use the same system and adversary model as in [6]. We assume a network that is composed of $N$ mobile nodes and a single offline Certification Authority run by an independent trusted third party. Mobile nodes are equipped with WiFi or Bluetooth-enabled devices and automatically exchange information upon meeting. Prior to entering the network, every mobile node registers with the CA that preloads a set of $M$ public/private key pairs and their digital certificates in the nodes.

We assume that the adversary aims to track the location of some nodes. We consider a passive adversary eavesdropping every messages in its coverage range. The adversary uses the pseudonyms sent along with messages to track the location of nodes. We assume that the adversary has an unknown coverage which in the worst case could cover the entire network. This type of adversary can be characterized as passive and global.

3 Pseudonym Change Games

The game studied in this paper is called the pseudonym change game and is defined in [6]. In this game, nodes meet in mix zones, and upon meeting must decide whether to change pseudonym or not. A node’s decision is based on its location privacy level and on its knowledge about other players and their location privacy levels.

The pseudonym change game is defined as a triplet $(P, S, U)$ where
• $\mathcal{P}$ is the set of players
• $\mathcal{S}$ is the set of strategies
• $\mathcal{U}$ is the set of payoff functions.

The set of players is the set of nodes meeting in the mix zone that are in the transmission range of each other. Upon meeting, each player has two possible moves $s_i$: Cooperate (C) or Defect (D). If a node decides to change a pseudonym we say that he cooperated, otherwise he defected. Formally, the set of strategies of each node $i$ where $i = 1, n$ (where $n$ denotes the number of players) is $S_i = \{C, D\}$.

The tricky part in the analysis of the approaches for providing location privacy is how to measure the level of location privacy nodes have and how to present the evolving of the level of location privacy in time. In [6], the authors introduced the user-centric location privacy model. In this model, the user-centric location privacy of a node $i$ at time $t$ is expressed with:

$$u_i(t) = u_i(T_i^l) - \beta_i(t, T_i^l)$$ (1)

where

• $u_i(T_i^l)$ is the level of location privacy achieved at the last successful pseudonym change of node $i$ at time $T_i^l$ (where at least one other node changed a pseudonym together with node $i$ in the mix zone)

• $\beta_i(t, T_i^l)$ is the location privacy loss function that is used to model the decrease of the level of location privacy in time.

According to [6] the location privacy loss function is equal to

$$\beta_i(t, T_i^l) = \begin{cases} \lambda(t - T_i^l) & : \text{for } T_i^l \leq t < T_i^l \\ u_i(T_i^l) & : \text{for } T_i^l \leq t \end{cases}$$ (2)

where $T_i^l$ is the time when the function reaches maximum value and $\lambda$ is the sensitivity parameter that models node’s beliefs of the tracking power of the adversary.

The achieved location privacy depends on both the node density and the unpredictability of node movements in mix zones. Like in [6], we assume that upon a pseudonym change, every node achieves the same level of privacy and thus we consider the upper-bound $\log_2(n_C)$ for the achieved location privacy which depends only on the number of nodes that changed pseudonyms in a mix zone $n_C$.

The payoff function of each node $i$ can be expressed as $u_i = b_i - c_i$ where $b_i$ is the benefit (the achieved level of location privacy) and $c_i$ is the cost (the cost of changing a pseudonym is denoted with $\gamma$; if a node does not change a pseudonym (plays D) the cost is 0). If a node $i$ plays $C$ and the number of other nodes that played $C$ is denoted with $n_C$, then the payoff of node $i$ can be expressed with

$$u_i(t, C) = \begin{cases} \max(0, u_i^m) - \gamma & : \text{when } n_C = 0 \\ \log_2(n_C) - \gamma & : \text{when } n_C > 0 \end{cases}$$ (3)

If a node $i$ plays $D$ then its payoff is equal to $u_i(t, D) = \max(0, u_i^-)$, where $u_i^-$ is the payoff node $i$ has at the time instant just before the game starts. It can be expressed with $u_i^- = u_i(T_i^l) - \beta_i(t, T_i^l) - \alpha_i(t, T_i^l)$, where $\alpha_i(t, T_i^l)$ is the counter of the number of wasted pseudonyms of node $i$ since its last successful pseudonym change at time $T_i^l$. We say that a player wasted a pseudonym if he played $C$ and all other players played $D$.

There are two types of players in the pseudonym change game: Friendly and Unfriendly. Formally, every player $i$ has a type $\theta_i \in \Theta$, where $i = 1..n$. The type of a player basically determines his probability for cooperation [6]. A Friendly player $i$ is a player whose utility $u_i$ is $u_i \leq \theta_T$, where $\theta_T$ is the threshold value to change a type. An Unfriendly player $i$ has a utility $u_i > \theta_T$. The rational behind these two types is that Friendly players have a low utility
so they will cooperate with the higher probability compared to Unfriendly players. Hence, the type of a player defines its behavior. On one hand, friendly players cooperate whenever \( u_i(t, C) > u_i(t, D) \), where \( u_i(t, C) \) and \( u_i(t, D) \) are the utilities of player \( i \) cooperating/defecting at time \( t \), defined above. On the other hand, unfriendly players cooperate only when \( u_i(t, C) > u_i(t, D) + r(\gamma) \), where \( r \) is a risk function defined as \( r(\gamma) = r\gamma \). The rational behind is that an unfriendly player will cooperate only if the privacy gain is larger than the potential loss \( r \) of a changing pseudonym alone.

4 Game analysis

In this report we analyze two types of dynamic games:

- dynamic games with complete information, and
- dynamic games with incomplete information.

In games with complete information, every player knows the location privacy level of the other players. This makes easier the prediction of the best move of other players. Since in realistic settings we cannot assume that nodes have the knowledge of their opponents’ location privacy level, we also study the games with incomplete information using a Bayesian approach [8]. Since in games with incomplete information, every player has incomplete information about its opponents utilities, Harsanyi [8] suggests the introduction of a new player named Nature that turns an incomplete game into an imperfect information game. To do so, Nature assigns a type to every player \( i \) according to pre-determined probability distribution \( q \). Because the type of a player essentially determines its behavior, every player can compute its optimal move based on its belief about the type of its opponent (i.e., Bayesian inference).

In dynamic games with incomplete information, players cannot obtain a Nash Equilibrium because they are not aware of the utilities of their opponents. Hence we introduce the notion of Bayesian Nash equilibrium [7],[8]. We use the notations and definitions of game-theoretic concepts from [6]. A pure strategy for a player \( i \) is a function \( s_i : \theta_i \to S_i \), where \( S_i = \{C, D\} \). The pure-strategy space is \( S_i^\theta \). We call the strategy profile \( s = \{s_i\}_{i=1}^{n} \) the set of strategies determining the move of every player.

**Definition 1.** A strategy profile \( s^* = \{s_i^*(.)\}_{i}^{n} \) is a pure-strategy Bayesian Nash equilibrium (BNE) if, for each player \( i \)

\[
\begin{align*}
\arg\max_{s_i(\cdot) \in S_i^\theta} \sum_{\theta_{-i}} q(\theta_{-i}) \cdot u_i(s_i(\cdot), s_{-i}(\theta_{-i}))
\end{align*}
\]

(4)

A mixed-strategy for player \( i \) is a function \( \sigma_i : \theta_i \to S_{im} \) where \( S_{im} \) specifies a mixed action for each type of player. The mixed strategy space is \( S_{im}^\theta \).

**Definition 2.** A strategy profile \( s^* = \{s_i^*(.)\}_{i}^{n} \) is a mixed-strategy Bayesian Nash equilibrium (BNE) if, for each player \( i \)

\[
\begin{align*}
\arg\max_{s_i(\cdot) \in S_i^\theta} \sum_{\theta_{-i}} q(\theta_{-i}) \cdot u_i(s_i(\cdot), \sigma_{-i}(\theta_{-i}))
\end{align*}
\]

(5)

We start with the analysis of 2-player Dynamic Game with Complete Information. Then we analyze the 2-player Dynamic Game with Incomplete Information. Then we extend the set of players to \( n \) and analyze \( n \)-player Dynamic Games with Complete and with Incomplete Information.

4.1 2-player Dynamic Game with Complete information

Let us consider the extensive form representation of the 2-player game (see Figure 1). The game is played once; one of the players \( P_1 \) makes his move first and then the other player makes his move. The users must maximize their payoff.

As we are assuming the game with complete information, each player knows the value of the payoff of its opponent: \( u_i(T^1_i), \beta(t, T^1_i), \alpha_i(t, T^1_i) \) and \( r \).
Lemma 1. The 2-player dynamic game has always three pure strategy Nash equilibria: \((C, CC),(C, CD)\) and \((D, DD)\).

Proof. \((C, CC)\) and \((C, CD)\) are Nash equilibria that leads to both players playing \(C\). Since achieved utility \(1 - \gamma\) in the 2-player game is always bigger than \(u_i - \gamma\) for \(i = 1, 2\), these two strategy profiles are Nash equilibria. Similarly \((D, DD)\) is an equilibrium because \(u_i > u_i - \gamma\) for \(i = 1, 2\).

Given Lemma 1 and the type of each player, the game will result in one of the three following scenarios:

- **Friendly vs Friendly**: As both users are cooperative, the NE strategy profile which is Pareto-optimal is preferable, i.e., \((C, CC)\) and \((C, CD)\).

- **Friendly vs Unfriendly**: Without loss of generality, let’s assume that \(P_1\) is friendly and \(P_2\) is unfriendly. Considering the extensive-form of the game (see Figure 1), the friendly node \(P_1\) will prefer to play \(C\), but unfriendly node \(P_2\) will decide to play \(C\) depending on its risk factor \(r_2\). In summary, the NEs \((C, CC)\) and \((C, CD)\) are selected if and only if \(\beta_2 + \alpha_2 > r_2\). Otherwise, the players prefer to play \((D, DD)\) NE.

- **Unfriendly vs Unfriendly**: The NEs \((C, CC)\) or \((C, CD)\) are selected if and only if \(\beta_i + \alpha_i > r_i\), \(\forall i\), otherwise \((D, DD)\) is preferred.

To summarize, if \(r \rightarrow 0\), then players always cooperate. Otherwise, the NEs \((C, CC)\) and \((C, CD)\) exists only when \(\beta_i + \alpha_i > r_i\), for \(i = 1, 2\). The larger the value of wasted pseudonyms \(\alpha_i\) is, the faster a player will turn friendly and start cooperating.

4.2 2-player dynamic game with Incomplete Information

In a game with incomplete information, the players know their own payoff functions, but do not know the type of their opponents. The type of their opponent is based on a priori knowledge provided by Nature, i.e. the probability distribution \(q\) of player types.
There are four possible encounters between two players: (F,F), (F,U), (U,F) and (U,U). We are assuming game in which one of the players, denoted $P_1$, plays its move first, and then a player $P_2$ plays its move. The player $P_1$ plays his best move based on his privacy level and based on a belief about type of the player $P_2$, while player $P_2$ observes the move of player $P_1$ and then plays his best move. The goal is to maximize the payoff function. We consider $P_1$ and analyze two possible scenarios: $P_1 = F$ and $P_1 = U$.

1) $P_1$ is Friendly: The game is represented as a tree, where the type of $P_2$ is chosen by Nature with probability $q$ (see Figure 2). Let us define $X_F = Pr\{u(t,C) > u(t,D)\}$ the probability of cooperation of a friendly node, and $X_U = Pr\{u(t,C) > u(t,D) + r\}$, the probability of cooperation of an unfriendly node.

In a mixed-strategy BNE, the player $P_1$ who plays the first chooses to cooperate according to probabilities $X_F$ and $X_U$ which are predefined.

**Lemma 2.** If $P_1$ is a friendly player, then he cooperates if and only if

\[ qX_F + (1-q)X_U > \frac{\gamma}{\beta_1 + \alpha_1 + \gamma} \]  \hspace{1cm} (6)

**Proof.** The expected utility of the player $P_1$ cooperating is

\[ E(u(t,C)) = q(X_F(1 - \gamma) + (1 - X_F)(u_1 - \gamma)) + (1 - q)(X_U(1 - \gamma) + (1 - X_U)(u_1 - \gamma)) \]  \hspace{1cm} (7)

The expected utility of the player $P_1$ defecting is

\[ E(u(t,D)) = u_1^- \]  \hspace{1cm} (8)

The player $P_1$ will cooperate if

\[ E(u(t,C)) > E(u(t,D)) \]

From this inequality we obtain:

\[ qX_F + (1-q)X_U > \frac{\gamma}{\beta_1 + \alpha_1 + \gamma} \]

\[ \Box \]
Lemma 3. If $P_1$ is an unfriendly player, then he cooperates if and only if
\[ qX_F + (1 - q)X_U > \frac{r_1 + \gamma}{\beta_1 + \alpha_1 + \gamma} \]  

Proof. When the player $P_1$ is unfriendly, he cooperates when
\[ E(u_1(t, C)) > E(u_1(t, D)) + r_1 \]
Applying similar reasoning as in proof of Lemma 2 we get
\[ qX_F + (1 - q)X_U > \frac{r_1 + \gamma}{\beta_1 + \alpha_1 + \gamma} \]

\[ \square \]

Lemma 4. Assume $P_1$ cooperates. If $P_2$ is
- friendly, then $P_2$ always cooperates,
- unfriendly, then $P_2$ cooperates if and only if $\beta_2 + \alpha_2 > r_2$, otherwise $P_2$ defects.

Assume $P_1$ defects, then $P_2$ always defects.

Let denote $x = qX_F + (1 - q)X_U$ the probability that one player cooperate, and $x_{T,F} = \frac{\gamma}{\beta_1 + \alpha_1 + \gamma}$ the threshold for a friendly player and $x_{T,U} = \frac{r_1 + \gamma}{\beta_1 + \alpha_1 + \gamma}$ threshold for an unfriendly player.

Then the strategy pair ($C$ if $x > x_{T,F}$, otherwise $D$, when $X_F$ is a probability of a cooperation of a friendly player and $X_U$ is a probability of a unfriendly player) is the mixed-strategy BNE when $P_1$ is friendly.

The mixed-strategy BNE for an unfriendly player $P_1$ is defined with the strategy pair ($C$ if $x > x_{T,U}$, otherwise $D$, when $X_F$ is a probability of a cooperation of a friendly player and $X_U$ is a probability of a unfriendly player)

Lemma 4 defines the best moves of the player $P_2$ (who plays the second) that corresponds to the BNE of the second player.

The obvious benefit in this game for a player is to plays second, because the second player have complete information because he can observe the move of the player who plays the first and hence, always plays his best move.

4.3 n-player Dynamic Game with Complete Information

In a n-player Dynamic Game with Complete Information, the set of players is extended from 2 to $n$. The players meet at time $t$ in a mix zone and take part in a pseudonym change game with complete information. Every player has a complete information about its opponent. The difference with the n-player Static Game with Complete Information is that players play their moves sequentially. But this difference does not influence the results related to possible Nash equilibria because players have complete information about each other and therefore can predict the best moves of each other. So the order in which moves are played does not affect results about possible Nash equilibria. This means that results from [6] related to the n-player Static Game with Complete Information apply also for the n-player dynamic games with complete information. For completeness we repeat it here.

Let us denote with $C_k$ a set of $k$ players cooperating and $\varepsilon$ the set of equilibrium of the game.

Lemma 5. The n-player dynamic game with complete information has $1 \leq |\varepsilon| \leq 2$ Nash equilibria and $\varepsilon = \{C^0D^n, \varepsilon_1\}$ where

\[ \varepsilon_1 = \left\{ \begin{array}{ll} C_kD^{n-k} & : \exists k, \text{ s.t. } \forall i \log_2(k) > A_i - \beta_i - \alpha_i \\ \emptyset & : \text{otherwise} \end{array} \right\} \]
4.4 n-player Dynamic Game with Incomplete Information

A set of players \( \{P_i\}_{i=1}^n \) meeting in a mix zone at time \( t \) take part in a n-player dynamic pseudonym change game with incomplete information. We assume a fixed ordering of the players, i.e. player \( P_1 \) is the player who plays the first, \( P_2 \) plays the second, \( P_i \) plays \( i-th \) and \( P_n \) plays its move the last.

1) Mixed-strategy BNE: Let denote with \( x \) the probability that one node cooperates, i.e. \( x = P_r(\text{"a node cooperates"}) = qX_F + (1 - q)X_U \). Let denote with \( x_k \) the probability that \( k \) nodes cooperate where
\[
x_k = P_r(\text{"k nodes cooperate"}) = \binom{n}{k} x^k (1-x)^{n-k}
\]

In the static game players meet in a mix zone and based on their beliefs about other players and its local level of location privacy simultaneously play their moves. In the n-player dynamic game, the order by which players play their moves is important, because player’s move is influenced by observed moves of players who already played their moves.

Let denote with \( n_i \) the number of nodes that played C before node \( P_i \) plays its move, \( n_i = 0..i - 1 \).

The expected utility of the player \( P_i \) cooperating is calculated by different formula depending on the value of \( n_i \). When \( n_i \) is equal to 0 the expected utility of cooperation of player \( P_i \) when \( i < n \) is equal to
\[
E(u_i(t, C)) = \sum_{k=0}^{n-i} x_k u_{ik}^+
\]
where
\[
u_{ik}^+ = \left\{ \begin{array}{ll} u_i^- - \gamma & : k = 0 \\
\log_2(k+1) - \gamma & : k > 0 \end{array} \right.
\]
When \( n_i \) is greater than 0 the expected utility of cooperation of player \( P_i \) when \( i < n \) is equal to
\[
E(u_i(t, C)) = \log_2(n_i + 1) + \sum_{k=1}^{n-i} x_k u_{ik}^+
\]
where
\[
u_{ik}^+ = \log_2(n_i + 1 + k) - \log_2(n_i + 1) - \gamma
\]
The expected utility of defection of player \( P_i \) is equal to
\[
E[u_i(t, D)] = u_i^-
\]
A friendly player \( i \) cooperates if and only if
\[
E[u_i(t, C)] > E[u_i(t, D)]
\]
An unfriendly player \( i \) cooperates if and only if
\[
E[u_i(t, C)] > E[u_i(t, D)] + r_i
\]
The player \( P_n \) who plays the last (\( i = n \)) plays his best move based on observed moves of the other players.
5 Dynamic ordering of players in n-player Dynamic Game with Incomplete Information

In this section we assume that the order of players in the n-player Dynamic Game is not fixed and not defined before the game takes place. In this section we present the algorithm for dynamic ordering of players in the n-player dynamic games with incomplete information (see Algorithm 1). The assumption we made is that the order of players in the n-player dynamic game is not fixed and not defined before the game takes place, and then we use Algorithm 1 to define the order in which the players will play their moves.

Initially, the order of all players is 1 (see line 2), and the number of players that played C is 0 (see line 3). The algorithm lasts maximally T (see lines 7 to 20) steps where the step lasts for the duration of one time unit. In every step, every player p calculates the expected values of cooperation using the equations 12 or 14 from the previous section using the current values of i_p and n_{ip} that are updated at the end of every step (see lines 26 to 28). Depending on its type (Friendly or Unfriendly), a player checks the conditions at line 14 if Unfriendly or condition at line 18 if Friendly. If the condition evaluates to true then the player enters the game and plays C. Otherwise, if the condition evaluates to false, then the player p update its order (i_p) and the number of players that already played C, and goes to the next step. In the following step, the player p will use observed moves of players that already played C to precisely calculate its expected value of cooperation. This can lead to the scenario in which a player whose the initial best move (in the first step of the algorithm), according to the calculated expected value and conditions at lines 14 or 18, was to defect, can change their mind in some of the following steps and decide to play C. Players, at which the conditions at lines 14 or 18 evaluates to false even in the last step (step T), play their moves at the end of the last step when they defect (see lines 23 to 25).

The advantage of the dynamic ordering presented in the Algorithm 1 is that it increases chance that the outcome of the game is equilibrium with the higher number of C moves played compared to the dynamic games with the fixed ordering, that leads to the higher achieved level of privacy for players. We will illustrate this advantage with the example. Let assume a 3-player game where players are denoted with P_1, P_2 and P_3. The levels of utility of players is such that:

- P_1 will cooperate no matter on moves of other players
- P_2 will cooperate if at least one player plays C before
- P_3 will cooperate if two players play C before.

In the dynamic games with the fixed ordering it can happen that the order is such that the player P_1 plays the first, the player P_3 the second and the player P_2 the third. This ordering leads to the CDD Bayesian Nash Equilibrium which does not lead to the increase of the utility for any player. With the dynamic ordering shown in the Algorithm 1, the player P_1 will play the first, then P_2 will play the second and P_3 the third. This leads to the CCC Bayesian Nash Equilibrium and all players will increase their utilities to log_2(3) − γ. This example illustrates that for the same setting dynamic ordering can give better result compared with the fixed ordering with respect to the average utility achieved.

This advantage of the dynamic ordering is also illustrated on Figure 3 for two different distribution of utilities of players (utilities are chosen according to the beta function β(a, b)). In the first case when the parameters of the beta functions are a = 2 and b = 2, the utilities are symmetric and centralized around 0.5 · (log_2(nMax) − γ). In the second case the parameters of beta functions are a = 2 and b = 5, the players have a small utility with high probability.

On Figure 3 we show how the average percentage of deviated decisions depends on the number of players that played the game for two different distributions of player’s utilities. By the deviated decision we denote the case when the in the first step of Algorithm 1 the best move was to defect and until the last step the player deviated from this decision and played C.

On Figure 3(b) we see that the average percentage of deviated decisions (for the case when the player’s utility are small with the high probability) is increasing with the increase in the number of players n, until n becomes equal to
Algorithm 1 Dynamic ordering of players in n-player Dynamic Game with Incomplete Information — code of player $p$

1: **Initialization:**
2: $i_p := 1$
3: $n_{ip} := 0$
4: $played_p := \text{false}$
5: $u_p$ /* current location privacy level */
6: $\theta, r, T$ /* $\theta$ is a threshold value that defines type of a player; $r$ is a value that represents risk; $T$ denotes duration of the game. All these values are predefined and the same for all players. */
7: for ($k := 1; k \leq T$; ) do
8:  if $played_p = \text{false}$ then
9:   if $n_{ip} = 0$ then
10:    $EC_i$ is calculated by formula (12)
11:  else
12:    $EC_i$ is calculated by formula (14)
13:  if $u_p \geq \theta$ then
14:    if $EC_i > u_p + r$ then
15:     PLAY ($C$)
16:     $played_p := \text{true}$
17:  else
18:    if $EC_i > u_p$ then
19:     PLAY ($C$)
20:     $played_p := \text{true}$
21:  wait until one time unit expires
22:  $k := k + 1$
23:  if $played_p = \text{false}$ then
24:   PLAY ($D$)
25:   $played_p := \text{true}$
26:  on $C$ played by some other player do
27:   $i_p := i_p + 1$
28:  $n_{ip} := n_{ip} + 1$

8. When $n$ is higher than 8 the average percentage of deviated decisions start decreasing. The reason is the following: for the values of $n$ higher than 7 the equilibrium achieved is all-$C$ (all players cooperate) and since the percentage of players whose initial decisions is $C$ increasing with the increase in the number of players (see Figure 4(b)) there is a smaller number of players that can change their initial decisions. Therefore the average percentage of deviated decisions is decreasing for values of $n$ that are higher than 8.

In the other case, when the parameters of the beta function are $a = 2$ and $b = 5$ the percentage of processes whose initial decision is $C$ is smaller than the case when $a = 2$ and $b = 2$ so there is a bigger space for processes to deviate from their initial decisions (see Figure 4(a)). Therefore the increase in the percentage of deviated decisions when the number of players is increasing (when $a = 2$ and $b = 2$) is higher compared to the other case when $a = 2$ and $b = 5$ (see Figure 3(a)).
Figure 3. The percentage of deviated decisions is increasing with the increase of the number of players that played the game when utilities are chosen according to $\beta(2, 2)$ distribution (see a)). When the utilities are chosen according to $\beta(2, 5)$ distribution (see b) the percentage of deviated decisions is increasing when $n \leq 8$ and then start decreasing for $n > 8$.

Figure 4. The average percentage of initial C-decisions for different number of players that played the game. On Figure a) player’s utilities are symmetric and centralized around $0.5 \cdot \log_2(nMax) - \gamma$. On Figure b) players have small utilities with high probability.

6 Simulations

In this section we show the results obtained by simulating n-player static and dynamic games with incomplete information in Matlab.

We choose the following parameters: $X_F = 0.5$, $X_U = 0.01$, $\theta_F = 0.3\log_2(nMax) - \gamma$, $nMax$ is the maximum number of players (in the simulations $nMax = 20$) and $\gamma = 0.2$. The values for $X_F$ and $X_U$ are chosen corresponding to the definitions of the friendly and unfriendly players, so that friendly players will cooperate with higher probability than unfriendly players. The player’s utilities are chosen randomly for each run according to the beta distribution $\beta(a, b)$ using the following formula $(\log_2(nMax) - \gamma) \cdot \beta(a, b)$ where $\log_2(nMax) - \gamma$ is the maximum location privacy level players can achieve if a maximum number of players is $nMax$. The results shown
on the graphs are averaged over 10000 simulations.

The beta distribution is a family of continuous probability distribution defined on the interval $[0, 1]$ and parameterized by two positive shape parameters $a$ and $b$. If $u_i \sim \beta(2, 5)$, nodes have small utilities with high probability (see Figure 5), whereas with $u_i \sim \beta(5, 2)$ nodes have large utilities with high probability (see Figure 6). If $u_i \sim \beta(2, 2)$, utilities are symmetric and centralized around 0.5 (see Figure 7). We assume that the distribution of utilities is known by the players so they can calculate $q$ with the following equation:

$$q = \int_0^1 f(U_i) dU_i$$

(18)

![Figure 5. The utilities of players randomly generated using $\beta(2, 5)$ distribution](image)

![Figure 6. The utilities of players randomly generated using $\beta(5, 2)$ distribution](image)

We use two metrics to evaluate the performances of different approaches:

- the average location privacy gain: if we denote with $u_i^-$ the level of privacy player $P_i$ ($i = 1..n$, where $n$ is the number of players) had before the game started and with $u_i^+$ the achieved level of privacy after the game is finished, then the average location privacy gain is equal to

$$\frac{1}{n} \sum_{i=1}^{n} (u_i^+ - u_i^-)$$

(19)
the average percentage of players that misused pseudonyms; by misused pseudonym we denote the case when a player $P_i$ played $C$ and the $u_i^- < u_i^+$, i.e. the pseudonym change didn’t lead to an increase of the level of location privacy.

Figure 8 illustrates the case when the utilities are distributed according to beta distribution with $a$ and $b$ parameters equal to 2 ($u_i^- \sim \beta(2, 2)$). We see from Figure 8(a) that the average location privacy gain is the best with dynamic games with dynamic ordering. This is the expected behavior because as shown in Section 5, for the same distribution
of utilities, the dynamic ordering provides an equilibrium with a higher number of cooperative players than with a fixed ordering. Players in static games achieve the worst results because they don’t have the possibility to observe the moves of other players since game is played simultaneously so they decide what to play only based on their belief of other players and on their local level of privacy.

In Figure 8(b), we see that the average percentage of players that misused their pseudonyms is smaller with both types of dynamic games, because players in dynamic games are able to precisely determine their best move by observing moves of other players. With this distribution of utilities, the average percentage of players that misused their pseudonyms for dynamic games with dynamic ordering is very similar to the static games and little bit higher than with dynamic games with fixed ordering. The reason can be the following: the distribution of utilities is such that there is a single player \( P_i \) whose utility is such that he plays \( C \) in static games and therefore plays \( C \) also in a dynamic games with the dynamic ordering. The utilities of other players are such that for smaller number of players they always defect. But if the order of player \( P_i \) is such that for example \( P_i \) plays 5th in a 10-player game, after observing 4 defections, the best move of \( P_i \) is to defect. Since for all three types of games, the percentage of cooperative players at equilibrium, i.e. the number of players that cooperated at equilibrium (see Figure 9), is pretty small when the number of players is smaller than 10, the example above show that a fixed ordering of players can reduce number of misused pseudonyms.

Still, the number of these scenarios is very small since at maximum value the average percentage of players that misused their pseudonyms is around 5 percent of total number of players that played the game.

Figure 9 illustrates how the average percentage of cooperative nodes at equilibrium evolves with the increase in the number of players in the case when both parameters of beta functions are 2. The reasoning similar as above can be used to show why for a small number of players \( (n < 10) \), the static games can have higher number of cooperative players at equilibrium than dynamic games with fixed ordering.

![Figure 9](image_url)

**Figure 9.** The average percentage of cooperative nodes at equilibrium is increasing with the increase in the number of players (utilities are chosen according to the beta distribution with parameters \( a = 2 \) and \( b = 2 \)).

Similar results are obtained for the case when \( u_i^- \sim \beta(2, 5) \) (see Figure 10). The order is the same as in the previous setting, but the achieved level of privacy is higher because in this setting there is a larger number of cooperative nodes at equilibrium (see Figure 11).

### 6.1 Comparing dynamic games with dynamic ordering with random strategy and social optimum

In this section we evaluate the efficiency of the BNE of the n-player dynamic games with incomplete information with dynamical ordering by comparing it with the random strategy and with the social optimum strategy. With the
random strategy nodes randomly decides their moves (whether to cooperate or defect). The social optimum strategy stands for the strategy in which players always cooperate. The simulation parameters are the same as in the previous section and the efficiency of the strategies is evaluated using the same metrics:

- the average location privacy gain, and
- the average percentage of players that misused pseudonyms.

The utilities of the players are chosen randomly according to the beta function with three different pairs of values for parameters $a$ and $b$:

- $a = 5, b = 2$
- $a = 2, b = 2$
Figure 12. When utilities are chosen according to $\beta(5, 2)$ distribution, the BNE of dynamic games with
dynamic ordering is all-D. Social optimum strategy achieves negative gain when $n < 10$ while random
strategy achieves negative gain for $n < 18$ (see Figure a)). The dynamic games with dynamic ordering
has the smallest number of misused pseudonyms (Figure b))

- $a = 2$, $b = 5$.

Figure 13. The comparison of dynamic games with dynamic ordering with random strategy and social
optimum strategy with respect to a) average location privacy gain and b) average percentage of players
that misused pseudonyms when the distribution of utilities is according to $\beta(2, 2)$ distribution

When the utilities of players are chosen according to $\beta(5, 2)$ distribution, the BNE of dynamic games with dynamic
ordering is all-D (which is a desired equilibrium because there is high percentage of players with high utilities), so the
achieved location privacy gain is 0 (see Figure 12). The social optimum strategy has a negative gain when number of
players is smaller than 10 and for this number of players very high percentage of misused pseudonyms. With $\beta(5, 2)$
distribution, the utilities of players are high (close to $\log_2(20)$) with the high probability, so when there is a small
number of players ($n < 10$), even when all nodes cooperate, the utility achieved is smaller than the utility players
Figure 14. The comparison of dynamic games with dynamic ordering with random strategy and social optimum strategy with respect to a) average location privacy gain and b) average percentage of players that misused pseudonyms when the distribution of utilities is according to beta function with parameters $a = 2$ and $b = 5$

have before the gain. This is the reason why for values of $< 10$, the average location privacy gain for the social optimum strategy is negative. The random strategy has a negative gain that is becoming smaller when the number of players is increasing and the percentage of misused pseudonyms is smaller than the for the social optimum strategy when number of players is $n < 10$. When the number of players becomes $n > 10$ then the social optimum strategy performs better than random strategy with respect to both metrics. The reason is that with the social optimum strategy all players cooperate, hence the achieved location privacy is higher than with the random strategy where the number of cooperative players is random.

For the other two distributions ($\beta(2,2)$ and $\beta(2,5)$), the location privacy gain achieved with dynamic games with dynamic ordering is similar to the gain achieved with the social optimum strategy (especially for $\beta(2,5)$, see Figure 14a)). But with respect to the percentage of misused pseudonyms, dynamic games with dynamic ordering has the better result than both other strategies (see Figure 13b) and Figure 14b)).

7 Non-cooperative Swing Protocol

In this section we show the complete changing pseudonym protocol for achieving the location privacy. It is designed using the knowledge gained from the game theory analysis which is applied in adaptation of the Swing protocol from [10].

In the original Swing protocol, a node performs update of a pseudonym only when changing its velocity, i.e. direction and speed. Therefore, the adversary can no longer use the tracking methods based on the predictable node movement (correlation tracking) [10].

In the Swing protocol, a node decides to change a pseudonym if it is not satisfied with its location privacy level and if in the neighborhood, there is at least one other node. In this case the node broadcasts an update message to trigger a pseudonym change. After the update message is broadcasted, the node stays silent for some random period of time and after this silent period, the node uses a new pseudonym.

In the new protocol called Non-cooperative Swing (see Algorithm 2), a node decides whether to change a pseudonym based on:

- its location privacy level
Algorithm 2 Non-cooperative Swing protocol — code of node $p$

1: **Initialization:**
2: $i_p := 1$
3: $n_{ip} := 0$
4: $played_p := \text{false}$  
5: $n_p := \{p\}$  /* set of players; initially contains only node $p$ */
6: $neighbours_p := 0$  /* set of neighbors; provided by a lower heartbeat service */
7: $u_p$  /* current location privacy level */
8: $\theta, r, s_{\text{min}}, s_{\text{max}}, s_{\text{max}}$  /* $\theta$ is a threshold value that defines type of a player; $r$ is a value that represent risk; $s_{\text{min}}$ and $s_{\text{max}}$ are time values that determine target reachable area; $s_{\text{max}}$ is the maximum speed. All these values are predefined and the same for all players. */
9: if changing velocity within $s_{\text{max}}$ then
   10: for ($t := 0; t <= s_{\text{max}}; t$)
   11: $n_p := n_p + neighbours_p + n_{ip}$
   12: if $played_p = \text{false}$ and not in silent period then
      13: if $n_{ip} = 0$ then
         14: $EC_i$ is calculated by formula (12)
      15: else
         16: $EC_i$ is calculated by formula (14)
      17: if $u_p >= \theta$ then
         18: if $EC_i > u_p + r$ then
            19: $played_p := \text{true}$
      20: else
         21: if $EC_i > u_p$ then
            22: $played_p := \text{true}$
         23: if $played_p = \text{true}$ then
            24: randomly choose silent period such that $s_{\text{min}} \leq \text{silent period} \leq s_{\text{max}}$
            25: broadcast UDP message $\langle \text{PID}_{i,i-1}, \text{"updating"} \rangle$ within distance $2s_{\text{max}}s_{\text{min}}$
            26: remain silent for silent period
            27: update $\text{PID}_{i,i-1}$ to $\text{PID}_{i,l}$
            28: wait until one time unit expires
            29: $t := t + 1$
      30: on received UDP message $\langle \text{PID}_{i,i-1}, \text{"updating"} \rangle$ from node $j$ do
         31: $i := i + 1$
         32: $n_{ip} := n_{ip} + j$

- size of its neighborhood
- the observed moves of the other players.

In other words, it incorporates strategic behavior. We assume that nodes are using the equations from Section 4.4 to measure its level of privacy. We also assume that nodes have a belief about the nodes they meet expressed through the values of $q$ (probability of friendly nodes), $X_F$ (probability of cooperation of friendly nodes) and $X_U$ (probability of cooperation of unfriendly nodes) that are necessary for calculating the expected utility of node by changing a pseudonym. The values of $q$, $X_F$ and $X_U$ in the real setting can be calculated by some heuristics based on a node’s experience with other nodes.

Some variables used in the Algorithm 2 are taken from the Swing protocol:

- a node can choose a speed in $[s_{\text{min}}, s_{\text{max}}]$
- a node can choose a length of a silent period in $[s_{\text{min}}, s_{\text{max}}]$. 

18
Algorithm 2 is started every time when a node changes its velocity within $s_{p_{\text{max}}}$ time units. It lasts maximally $(s_{p_{\text{max}}} + 1)$ steps (one step presents one time unit) where in each step the expected value of a cooperation is calculated using the updated values of

- its local location privacy level
- the number of nodes that changed their pseudonym
- the number of neighbors.

This part of the protocol is taken from Algorithm 1.

The number of nodes that changed their pseudonyms is calculated from the moment node starts the protocol instance (from the moment when the node changes its velocity within $s_{p_{\text{max}}}$ time units). If in some of the $(s_{p_{\text{max}}} + 1)$ steps the expected value of cooperation of a node is such that the changing a pseudonym is the best move (according to the game theory analysis) then the node

- broadcasts the update message within distance $2s_{\text{max}}s_{p_{\text{max}}}$ that it is entering update,
- stays silent for a random period between $s_{p_{\text{min}}}$ and $s_{p_{\text{max}}}$ time units and
- update the pseudonym.

The actions of a node after deciding to change its pseudonym are the same as in the original Swing protocol.

![Graph (a) average location privacy gain](image1)

![Graph (b) average percentage of players that misused pseudonyms](image2)

**Figure 15. The comparison of Non-cooperative Swing protocol with the original Swing protocol with respect to a) average location privacy gain and b) average percentage of players that misused pseudonyms when the distribution of utilities is according to beta function with parameters $a = 2$ and $b = 2$.**

We simulate in Matlab a simplified version of the Swing protocol and of the Non-cooperative Swing protocol where nodes are static. The comparison is done with respect to the same metrics used in the previous section and utilities are chosen according to beta distribution for two values of parameters $a$ and $b$: (i) $a = 2$, $b = 2$ and (ii) $a = 2$, $b = 5$.

The results are shown on Figure 15 and Figure 16 where it is shown that Non-cooperative Swing protocol outperforms the original Swing protocol with respect to both metrics. In the simulated Swing protocol, players cooperate if their location privacy level is below threshold (if they are friendly with respect to the definition of Friendly player type in Pseudonym Change Game). So the percentage of cooperative players depends only on the number of players whose utilities are below threshold. Hence, when the percentage of players whose utilities are below the threshold is
Figure 16. The comparison of Non-cooperative Swing protocol with the original Swing protocol with respect to a) average location privacy gain and b) average percentage of players that misused pseudonyms when the distribution of utilities is according to beta function with parameters $a = 2$ and $b = 5$.

not high (which is the case for example when the utilities are chosen according to $\beta(2, 2)$ distribution), the average location privacy gain is small (see Figure 15(b)). This also leads to the high percentage of misused pseudonyms (see Figure 15(b)). On the other side, players with the Non-cooperative Swing protocol are more strategic in deciding when to change a pseudonym (they consider their location privacy level, size of their neighborhood and the observed moves of other players). Therefore, the average location privacy gain with the Non-cooperative Swing protocol is much higher than with the Swing protocol. This also leads to the much smaller percentage of misused pseudonyms.

8 Conclusion

The work presented in this report is an extension of the work done in [6]. We use the user-centric location privacy model defined in [6] to analyze changing pseudonym approach for achieving location privacy in non-cooperative setting using the game theory. In [6], authors defined the pseudonym change game and analyze the non-cooperative behavior of mobile nodes in the context of static games with complete and incomplete information. In this report, we analyze the pseudonym change game in the context of dynamic games and obtained NE strategy profiles for dynamic games with complete information and BNE strategy profiles for dynamic games with incomplete information. In the analysis of dynamic games with incomplete information we use a little bit different model than the model used in [6]. We have assumed that players have predefined values $q, X_U$ and $X_F$, while in [6] they assumed that only predefined information is distribution of player’s utilities. Therefore, the BNE strategy profiles obtained for dynamic games with incomplete information are different from the results in [6]. It is an open question, which of these two model will have better results in practical settings.

The efficiency of different type of games are compared using simulations in Matlab. We use two metrics to evaluate the efficiency of different games: (i) the average location privacy gain and (ii) the average percentage of players that misused pseudonyms. In the context of dynamic games with incomplete information we analyzed two different approaches: 1) dynamic games with fixed ordering and 1) dynamic games with dynamic ordering. We design the new algorithm for dynamic ordering of players in dynamic games with incomplete information. The dynamic games with dynamic ordering has the best result with respect to the both metrics. We also compare the BNE strategy for dynamic games with dynamic ordering with random strategy and social optimum strategy for different distribution of player’s utilities. The BNE strategy profile achieves similar the average location privacy gain as social optimum strategy but
has a significantly smaller percentage of players that misused pseudonyms.

Using the knowledge gained from game theory analysis, we design a new complete changing pseudonym protocol called Non-cooperative Swing, that is the extension of the Swing protocol [10] for non-cooperative settings. We compared the efficiency of simplified version of the original Swing protocol with the new protocol and shown that the Non-cooperative Swing achieves significantly better results. The simulation of these two protocols in some of the network simulators that can simulate the movement of nodes is left for some future work.

References


