I. J. Computer Network and Information Security, **2016, 11, 1-13**
Published Online November 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.11.01

Security Requirements Metrics for Pattern-Lock Applications on Mobile Devices

Irfan Afifullah, Bayu Hendradjaya
School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia
E-mail: 23513170@std.stei.itb.ac.id, bayu@stei.itb.ac.id

Abstract—Pattern-Lock is one of graphical authentication schemes that shows high popularity today. Based on recent research, the security requirements metrics of Pattern-Lock applications have not proposed yet. The goal of this study is to define security requirements metrics for Pattern-Lock applications on mobile devices. Our study has identified 12 threat statements and 18 requirements statements by analyzing STRIDE (Spoofing the identity, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) and Extended Misuse Case diagram. To develop the metrics we have used Goal-Question-Metric (GQM) paradigm. Based on these, we develop 3 Goals and 7 Questions and resulted in 20 metrics for security requirements. The metrics have been evaluated using 30 App Locker Android applications, and the results show that some metrics have higher values than others. Number of Pattern Characteristics that Successfully Detected, Ability to Relock, and Grid Size metrics have the three highest values. These metrics requires higher priorities to look into when developers need to build the App Locker applications. Moreover, developers should ensure that App Locker applications have values higher than average of security goals and metrics achievements.

Index Terms—GQM, Pattern-Lock, Requirement Statements, Security Requirements Metrics, Threat Statements.

I. Introduction

Touchscreen technology has triggered the implementation of graphical authentications on smartphones. One example of graphical authentication schemes that has high popularity today is Pattern-Lock. Pattern-Lock is a scheme that utilized 3x3 grid and usually numbered from 1 (upper left corner) to 9 (lower right corner) like in [1]. Based on a classification of graphical authentication schemes from [2], i.e. Drawmetric, Locimetric, Cognometric, and Hybrid, the Pattern-Lock scheme can be classified as Drawmetric class because user creates a password by drawing a pattern.

Pattern-Lock scheme on a smartphone can be implemented as a privacy protector application, either Screen Locker or App Locker. Screen Locker is used to lock screen and App Locker is used to lock private or confidential applications. Android as one of popular operating systems on smartphones has a Screen Locker type. However, users can also download the Pattern-Lock applications from Google Play Store.

Like in [3], graphical authentication, including Pattern-Lock, at least has two advantages compared to textual authentication. First, graphical authentication is relative easier to use without needs to create complex alphanumeric combinations. Second, graphical authentication is more fun to use because more entertaining by creating a password graphically. However, there is a dilemma between security, usability, and convenience according to [2], [4], [5]. If easier and more convenient to use, the password tends to be less secure. And vice versa. Furthermore, easily guessable password, user, and bug in software application are the weakest link in the information security domain [4]. Therefore, security has been a major concern for Pattern-Lock.

To strengthen the security of Pattern-Lock, researchers have proposed metrics, e.g. pattern length [3], [6], [7], line visibility [7], knight moves [7], intersecting points [6], [7], overlapping points [7], entropy [3], [8], start and end points [3], sub-pattern analysis [3], and ratio of non-repeated segments [6]. These research only discuss security for Pattern-Lock scheme, not Pattern-Lock applications. Therefore, to the best our knowledge, metrics for Pattern-Lock applications have not proposed yet. As more users to use Pattern-Lock applications to lock screen or applications, so that there is a need to measure security requirements for Pattern-Lock applications.

The research question of this study is how to define security requirements metrics of Pattern-Lock applications on mobile devices. The goal of this study is to define security requirements metrics using Goal-Question-Metric (GQM) paradigm from [9]. To achieve that goal, this study has objectives to identify the threat model and specify security requirements. This study contributes to propose Threat Statements (TS), Requirement Statements (RS), Metrics (M) of Pattern-Lock applications, and evaluate the proposed metrics on 30 App Locker applications on Android. The results can be used by Pattern-Lock application developers as a security requirements checklist and by users as a tool to help choose Pattern-Lock applications.

The rest of this paper is organized as follows. Section II provides literature review. Section III describes research methodology. Section IV gives activities for evaluating the metrics, data collection preparation, and

data collection results. Section V discusses proposed metrics, data collection results, and threats of validation. Finally, Section VI presents conclusions and future work.

II. Literature Review

Recent research about Pattern-Lock can be roughly divided into three topics, i.e. pattern strength, specific threats for Pattern-Lock, and forensic.

A. Pattern Strength Topic

Reference [6] measure pattern strength by utilizing 3 pattern characteristics, i.e. pattern length, ratio of non-repeated segments, and number of intersecting points. Based on one-tailed unpaired t-test for implementation of the Pattern-Lock strength meter, pattern length and number of intersecting points have statistically significant differences, i.e. $p < 0.01$ and $p < 0.05$ respectively. However, ratio of non-repeated segments is not successful to show a statistically significant difference because has $p = 0.458$.

Reference [3] measure pattern strength by utilizing 4 pattern characteristics, i.e. pattern length and direction changes, entropy, start and end points, and sub-pattern analysis. The results are most collected patterns contain 5-7 points, 51.7% patterns are starting from upper left corner, and number of patterns that labeled to "Weak", "Medium", and "Strong" are 32, 44, and 44 respectively after suggested the users to change the pattern.

Reference [7] measure strength from 5,960 patterns by utilizing 5 pattern characteristics, i.e. pattern length, line visibility, knight moves, intersecting points, and overlapping points. In general, 51.7% (3,565 patterns) can be guessed only in one observation where lines from 57.9% of 3,565 patterns are visible. If the line is invisible, pattern length is increased by 1 point, and adds knight move, intersecting point, overlapping point, the percentage can be reduced by 67%, 45%, 32%, 12%, and 20% respectively.

Reference [8] through the user study with 584 participants investigate actual entropy. This investigation found that the pattern selection process in a Pattern-Lock scheme has high bias, e.g. upper left corner and straight line through three points.

In contrast to the above research, this study aims to investigate the security of Pattern-Lock applications rather than Pattern-Lock scheme by defining security requirements metrics. The pattern characteristics and results from the above research are useful to develop the proposed metrics.

B. Specific Threats for Pattern-Lock Topic

This study found five specific threats from literature that possible for Pattern-Lock, i.e. smudge attack, shoulder surfing, guessing, tricking, and spyware.

Smudge Attack: Reference [1] use photography technique through three conditions, i.e. ideal collection, simulated usage, and removing smudges, showed that smudge attack is feasible for Android Pattern. Reference [10] through the user study with 24 participants propose three graphical authentication schemes to improve existing Android Pattern, i.e. Pattern 90, Marbles, and Marble Gap. Based on quantitative and qualitative results, Marble Gap is the most secure scheme against smudge attack followed by Marbles, Pattern 90, and Android Pattern.

Shoulder Surfing: First experiment from [6] found that the patterns that labeled to "Strong" are more resistant than "Medium" and "Weak" ones. Reference [7] also evaluate the shoulder surfing and found that line visibility and pattern length are the most important factors for successful shoulder surfing attack. Reference [2], [11] also state that shoulder surfing as one of the threats for graphical authentication.

Guessing: Second experiment from [6] found that 10% patterns that generated without and with Pattern-Lock strength meter can be guessed after 16 and 48 attempts respectively. Reference [8] measure the partial entropy at 3 levels, i.e. 10%, 20%, and 50%. At level 20% obtained that the partial guessing entropy is 9.10 bits. This value is lower than random 3-digit PIN (Personal Identification Number), i.e. 9.97 bits. After the layout of the Pattern-Lock scheme is changed, the partial guessing entropy is increased to 10.81 bits. Reference [11] also state that guess password as one of the threats for graphical authentication.

Tricking: Reference [2] state Tricking as one of the threats for graphical authentication as a social engineering attack. In Tricking, Misuser can tricks the users to find the correct pattern. Tricking is similar with Password Revealment in [11]. In Password Revealment, users can share the patterns with others although more difficult than textual authentication.

Spyware: Reference [2], [11] also state that spyware can also become a threat for Pattern-Lock. Reference [2] provide three types of spyware, i.e. keystroke-loggers, mouse-loggers, and screen-scrapers. Pattern-Lock is more vulnerable with screen-scrapers because the patterns can be captured.

In contrast to the above research, this study focuses to model the threats of Pattern-Lock applications as a basis for security requirements metrics rather than deeply analyses certain threats. Names of threats from the above research are useful to identify the threats of Pattern-Lock applications.

C. Forensic Topic

Reference [12] propose a data acquisition method for Android and applied on 6 Android-based smartphones using 4 scenarios. Pattern-Lock scheme is discussed in Scenario I, i.e. Turned on: No, Removable card: Yes, Locked: Yes, Unlockable: Yes, and Super user: No. Users can install the Screen Lock Bypass application to unlock screen by using ADB (Android Debug Bridge). Reference [13] utilize forensic guidance from NIST (National Institute of Standards and Technology) to do physical acquisition, create image file from the database, and unlock the pattern when Android-based smartphone is locked by the pattern.

In contrast to the above research, this study finds the

pattern to change from lock to unlock condition on an Android-based smartphone without forensic process, e.g. device is on, not use forensic tools, and screen can be unlocked or locked because the targets on this study are locked applications.

III. Method

The methodology that used in this study are literature study, metrics development (threat modeling, security requirements definition, and security requirements metrics definition), and metrics evaluation. The literature study, threat modeling, and security requirements definition can be mapped to a requirement engineering process from [14]. On the other hand, the security requirements metrics definition and metrics evaluation can be mapped to a measurement process from Roche (1995) in [15]. The literature study has described in Section II. Below are explanation for metrics development and metrics evaluation.

A. Metrics Development

This stage contains three activities, i.e. threat modeling, security requirements definition, and security requirements metrics definition.

Threat Modeling: The steps for threat modeling in this study are Application Modeling, Threats Identification, Vulnerabilities Identification, and Threat Model Validation. These steps are adapted from four-stage framework [16], i.e. Model System, Find Threats, Address Threats, and Validate.

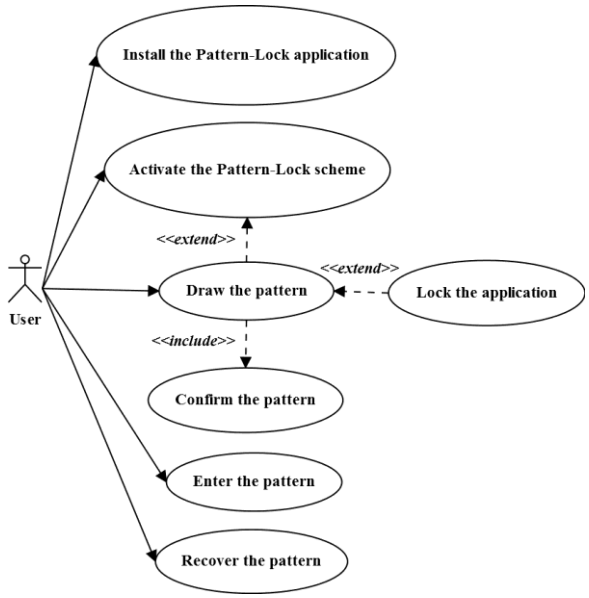

Fig.1. Use Case Diagram for App Locker Applications

The first step (Application Modeling) utilized Use Case diagram. The Application Modeling has identified four common use cases for App Locker applications, i.e. install the Pattern-Lock application, create the pattern, enter the pattern, and recover the pattern. After users create the pattern, the users must confirm the pattern and then can lock the private or confidential applications. In some Pattern-Lock applications, users can activate the Pattern-Lock scheme first before deciding to create the pattern. The Use Case diagram for App Locker applications is shown in Fig. 1.

The second step (Threats Identification) provides Threat Statements (TS) that utilized mnemonic STRIDE (Spoofing the identity, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege). Based on [16], STRIDE was invented by Loren Kohnfelder and Praerit Garg in 1999. Each element of STRIDE can compromises security properties, i.e. Authentication, Integrity, Nonrepudiation, Confidentiality, Availability, and Authorization respectively. The proposed threat statements for Pattern-Lock applications on mobile devices are shown in Table 1.

Table 1. Proposed Threat Statements (TS) for Pattern-Lock Applications on Mobile Devices

Label	Name	Description
TS1	Smudge Attack	Misuser attempts to get privileged access as a user by entering the pattern through a smudge on the smartphone screen (*E*).
TS2	Shoulder Surfing	Misuser observes the pattern or pattern recovery when entered by user (*I*).
TS3	Guessing	Misuser attempts to get privileged access as a user by guessing the pattern or pattern recovery (*E*).
TS4	Tricking	Misuser gets information about pattern or pattern recovery directly from the user. For example, Misuser can use psychological approach to get privileged access as a user (*E*).
TS5	Spyware	Misuser installs spyware that have capability to capture screenshots on the user's smartphone to obtain a piece of pattern, pattern recovery, and private or confidential information (*I*).
TS6	Uninstalling	Misuser deletes the Pattern-Lock application (*T*), so that pattern created by the user also deleted automatically.
TS7	Log Deleting	Misuser deletes the log (*T*), so that Misuser can denies never enter the pattern (*R*).
TS8	Force Stopping	Misuser forces to stop the Pattern-Lock application (*D*).
TS9	Authentication Bypassing	Misuser bypasses certain authentication steps (*E*).
TS10	Crashing	Misuser makes the Pattern-Lock application becomes crash or work abnormally (*D*).
TS11	Lock Inconsistency Attack	Misuser exploits the lock inconsistency by opening the locked application several times without entering the pattern or pattern recovery to see locked application content for a while (*I*).
TS12	Relock Timeout Misusing	Misuser takes benefit from the time interval before relock timeout to enter into the locked application (*I*).

TS1-TS5 are based on literature study, i.e. Smudge Attack [1], [10], Shoulder Surfing [2], [6], [7], [11], Guessing [3], [6], [8], [11], Tricking [2] which similar with Password Revealment [11], and Spyware [2], [11]. On the other hand, TS6-TS12 are based on analysis using

Pattern-Lock applications on Android directly and users reviews checking on Google Play Store.

The third step (Vulnerabilities Identification) has identified three vulnerabilities for Pattern-Lock applications on mobile devices that labeled from V1 to V3, i.e. install the Pattern-Lock application, enter the pattern, and recover the pattern. The interesting finding that installing the Pattern-Lock application is vulnerable because Pattern-Lock application has bugs that can reduce the reputation as a privacy protector. The identified vulnerabilities then was paired with the identified threats based on Threat Statements (TS). Threat-vulnerability pairs are shown in Table 2.

Table 2. Threat-Vulnerability Pairs

Threat based on Threat Statements (TS)	Vulnerability (V)
TS6, TS7, TS8, TS10, TS11, TS12	V1: Install the Pattern-Lock application
TS1, TS2, TS3, TS4, TS5, TS9	V2: Enter the pattern
TS2, TS3, TS4, TS5, TS9	V3: Recover the pattern.

The identified threats and vulnerabilities then was illustrated using Extended Misuse Case diagram from [17]. The Extended Misuse Case diagram extends Misuse Case diagram from [18]. The Extended Misuse Case diagram was chosen because the diagram has threat and vulnerability notations. The Extended Misuse Case diagram should separates outside and inside attackers. However, to investigate the Pattern-Lock applications, this study does not separate outside and inside attackers because both of them can misuse the Pattern-Lock applications. Therefore, this study still uses Misuser as stated on Misuse Case diagram to both outside and inside attackers.

The fourth step (Threat Model Validation) provides the validation of proposed threat statements, i.e. completeness and existency checking as follows:

- The completeness checking was done by mapping threat statements and STRIDE as shown in Table 3. The completeness checking showed that all proposed threat statements have mapped to STRIDE, except for the S element of STRIDE, i.e. spoofing the identity. This is due to this study does not investigate the fake Pattern-Lock applications.

Table 3. Mapping between threat statements and STRIDE

STRIDE	Threat Statements (TS)	Security Property
S	-	Authentication
T	TS6, TS7	Integrity
R	TS7	Nonrepudiation
I	TS2, TS5, TS11. TS12	Confidentiality
D	TS8, TS10	Availability
E	TS1, TS3, TS4, T9	Authorization

- The existency checking was done by checking proposed threats directly or indirectly. All proposed threats are directly checked, except for Smudge Attack (TS1), Shoulder Surfing (TS2), and Tricking (TS4). This is due to more appropriate if investigated by conducting user study, like in [10] for smudge attack and [6], [8] for shoulder surfing. After checking the threats, the misuse case narration from [18] then utilized for each proposed threat. An example of misuse case narration is shown in Table 4 for Lock Inconsistency Attack. The existency checking showed that all proposed threats exist.

Table 4. Misuse case narration for Lock Inconsistency Attack

Field	Description
Misuse Case Name	Lock Inconsistency Attack.
Summary	Misuser takes benefit from lock inconsistency of Pattern-Lock application.
Basic Path	1. Misuser take user's smartphone physically. 2. Misuser open the locked application several times until lock inconsistency appears.
Alternative Paths	-
Mitigation Points	1. Misuser failed to take user's smartphone because user is very protective. 2. Misuser failed to find the Pattern-Lock application. 3. Misuser failed to find the locked application.
Extension Points	-
Preconditions	1. User neglects the smartphone. 2. Pattern-Lock application is not hidden. 3. Locked application is not hidden.
Postcondition	Misuser success to enter in the locked application.

Based on three approaches for threat modeling from [16], i.e. asset-centric, software-centric, and attacker-centric, this study tends to use software-centric by utilizing Use Case diagram and attacker-centric by utilizing misuse case narration.

Security Requirements Definition: Reference [19] conclude that threat modeling can be used to specify security requirements. Therefore, this study also utilized threat modeling to specify security requirements. This study adopts a requirement engineering process from [14], i.e. Feasibility Study, Requirement Elicitation and Analysis, Requirement Specification, and Requirement Validation to specify the security requirements. The first step (Feasibility Study) utilizes literature review. Section II showed that the security requirements of Pattern-Lock applications are feasible. The second step (Requirement Elicitation and Analysis) was done by threat modeling activity as given in previous explanation.

The third step (Requirement Specification) provides Requirement Statements (RS), both global and specific ones. The global security requirement (RS0) is "The application must provide enough security features to secure pattern, pattern recovery, log, Pattern-Lock application, and locked applications from Misuser". Based on [20], proposed security requirements should be specified by security objectives. The security properties of identified STRIDE elements are utilized as security objectives. This study has specified 18 specific security requirements statements for Pattern-Lock applications on mobile devices as shown in Table 5. Some of specific security requirements can be mapped to the security aspects of textual authentication from [4], i.e. RS1 (Strong Password, Password Filtering, Pronounceable

Password), RS2 (Limited Login Attempts), RS3 (Artificial Delay), RS7 (Image Authentication), RS8 (Password Encryption), and RS18 (Last Login).

Table 5. Proposed Requirement Statements (RS) for Pattern-Lock applications on mobile devices

Label	Description	Security Property
RS1	The application shall detect the pattern strength.	Authorization
RS2	The application shall implement limited attempts for wrong pattern.	Authorization
RS3	The application shall implement the interval time that increase gradually after wrong pattern entered.	Authorization
RS4	The application shall implement the different patterns to lock different applications.	Authorization
RS5	The application shall implement the remote locking.	Authorization
RS6	The application shall prevent the pattern recovery misusing.	Authorization
RS7	The application shall provide the additional security mechanism to Pattern-Lock.	Authorization
RS8	The application shall store the pattern in non-plain text format.	Confidentiality
RS9	The application shall ensure confidentiality after user exits from locked application.	Confidentiality
RS10	The application shall hide the pattern.	Confidentiality
RS11	The application shall hide the pattern recovery.	Confidentiality
RS12	The application shall hide the Pattern-Lock application.	Confidentiality
RS13	The application shall hide the locked application.	Confidentiality
RS14	The application shall ensure lock consistency.	Confidentiality
RS15	The application shall prevent the Pattern-Lock application deleting.	Integrity
RS16	The application shall prevent the log deleting.	Integrity
RS17	The application shall resist from attempts that can make application becomes crash or work abnormally.	Availability
RS18	The application shall record successful and failed attempts.	Nonrepudiation

The fourth step (Requirement Validation) provides the validation of proposed security requirements, i.e. completeness and realism checking as follows:

- The completeness checking was done by mapping requirement statements and threat statements as shown in Fig.2. The completeness checking showed that all proposed requirements statements have mapped to proposed threat statements. For example, strong pattern (RS1) can makes Misuser longer to the find correct pattern through Smudge Attack (TS1). Furthermore, strong pattern can be also makes Misuser difficult to find the correct pattern through Shoulder Surfing (TS2), Tricking (TS4), or Spyware (TS5).

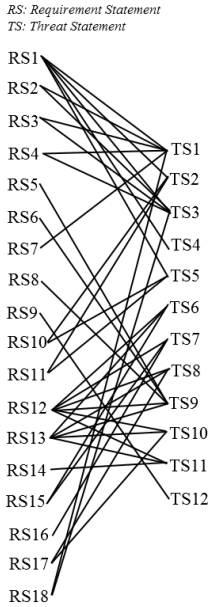

Fig.2. Mapping between Requirement Statement and Threat Statement

- The realism checking was checked by considering S(ecurity)-U(sability)-C(ost) Pyramid from [5]. The realism checking showed that proposed requirement statements have S-U-C scores as follows: security (high: 18 of 18, low: 0 of 18), usability (high: 6 of 18, low: 12 of 18), and cost (high: 8 of 18, low: 10 of 18) as shown in Table 6. Therefore, the proposed requirement statements have high security, low usability, and low cost. For example, RS1 can drives users to make a strong pattern (high security), but users will be less convenient (low usability). The Pattern-Lock applications shall be identifying strong pattern characteristics, e.g. intersecting and overlapping points. The Pattern-Lock application developers will take more time and energy to implement and test the pattern characteristics (high cost).

Table 6. Requirement Statements Checking Against Security, Usability, and Cost

Requirement Statements (RS)	Security	Usability	Cost
RS1	High	Low	High
RS2	High	Low	Low
RS3	High	Low	Low
RS4	High	Low	High
RS5	High	Low	High
RS6	High	Low	Low
RS7	High	Low	High
RS8	High	High	Low
RS9	High	High	Low
RS10	High	Low	Low
RS11	High	Low	Low
RS12	High	Low	Low
RS13	High	Low	Low
RS14	High	High	High
RS15	High	High	High
RS16	High	Low	Low
RS17	High	High	High
RS18	High	High	High
High:	18	6	8
Low:	0	12	10

Security Requirements Metrics Definition: This activity can be mapped to Formulation phase based on a measurement process from Roche (1995) in [15]. This study utilized Goal-Question-Metric (GQM) paradigm from [9]. Previous research that also utilized GQM for security requirements, e.g. [21], [22].

This study specified global and specific issues. The global issue is security and the specific issues are authorization, confidentiality, integrity, availability, and nonrepudiation based on identified security properties from the threat modeling activity. This study utilized GQM template as follows:

- G1. [*Purpose*] Ensure and improve [*Object*] the Pattern-Lock applications on mobile devices for aspect [*Issue*] authorization from [*Viewpoint*] user.
- G2. [*Purpose*] Ensure and improve [*Object*] the Pattern-Lock applications on mobile devices for aspects [*Issue*] confidentiality, integrity, and availability from [*Viewpoint*] user.
- G3. [*Purpose*] Ensure and improve [*Object*] the Pattern-Lock applications on mobile devices for aspect [*Issue*] nonrepudiation from [*Viewpoint*] user.

After Security Issues and Goals (G) have defined, then Questions (Q) and Metrics (M) are defined. This study has identified 3 Goals, 7 Questions, and 20 Metrics of Pattern-Lock applications on mobile devices. The global view of Goals, Questions, and Metrics are shown in Table 7. In the detail view, each proposed metric contains the rationale, threshold, possible values, interpretation, and test procedure as follows:

- Rationale, e.g. M1: if grid size is bigger, the user can creates more complex patterns.
- Threshold, i.e. M1: 3x3, M2: 4/9, M3: 1, M7: Increase, M16: Non-plain Text, M20: 3, M4, M5, M6, M8, M9, M10, M11, M12, M13, M14, M15, M17, M18, M19: Capable.
- Possible values, i.e. using qualitative nominal (M4, M5, M6, M8, M9, M11, M12, M13, M14, M17, M18: Capable, Not Capable, Not Applicable; M10, M15, M19: Capable, Not Capable; M16: Non-plain Text, Plain Text, Not Applicable), qualitative ordinal (M7: Constant, Increase, Not Applicable), quantitative ordinal (M1: <3x3, 3x3), quantitative ratio (M2: 1/9, 2/9, 3/9, 4/9), and quantitative absolute (M3: 1, 2, 3, …; M20: 0, 1, 2, 3, ...).
- Interpretation, i.e. SECURE if the value is equal or higher than the threshold, LESS SECURE if the value is lower than the threshold, or NA (Not Applicable) if the metrics are not applicable because the intended features are not available.
- Test procedure, e.g. M1: to check grid size is try to change size and choose the smallest size because has more impact for pattern strength.

Table 7. Proposed security requirements metrics for Pattern-Lock applications on mobile devices

Goal (G)	Question (Q)	Metric (M)
G1. Authorization	Q1. How to ensure that the pattern is strong?	M1. Grid size.
		M2. Ratio between minimum dots and grid size.
		M3. Number of pattern characteristics that successfully detected
	Q2. How to ensure from unauthorized access?	M4. Ability to use different patterns to lock different applications.
		M5. Ability to lock remotely.
		M6. Ability to limit failed attempts.
		M7. Interval time of artificial delay.
		M8. Ability to provide the additional security mechanism.
	Q3. How to ensure from misuse of pattern recovery?	M9. Ability to prevent the misuse of pattern recovery.
G2. Confidentiality, Integrity, and Availability	Q4. How to ensure that pattern, pattern recovery, Pattern-Lock application, locked application, content, and data are confidential?	M10. Ability to relock.
		M11. Ability to hide the pattern when entered.
		M12. Ability to hide the pattern recovery when entered.
		M13. Ability to hide the Pattern-Lock application.
		M14. Ability to hide the locked application.
		M15. Ability to keep locking consistently.
		M16. Form of data storage.
	Q5. How to ensure preventing deletion Pattern-Lock application and log?	M17. Ability to prevent the Pattern-Lock application deleting.
		M18. Ability to prevent the log deleting.
	Q6. How to ensure availability of Pattern-Lock and locked applications?	M19. Ability to resist from interruption.
G3. Nonrepudiation	Q7. How to ensure Misuser identification?	M20. Number of log information.

In [15] there are differences between Measure, Measurement, Metric, and Indicator. Furthermore, [23] differentiate between Measurement and Calculation. In the context of this study, the "Size" in M1 and the "Ratio" in M2 are Measures. The "Grid Size" and the "Ratio between minimum dots and grid size" are Metrics. The activity to obtain value M1 is Measurement, but Calculation for M2 because combines the results from number of minimum dots and grid size. This study prefers Test Procedure to Measurement or Calculation. The users still create a weak pattern, so that Misuser easy to guess that pattern is Indicator for M1 and M2.

Metric validation techniques in the literature are for assessment or prediction purpose [23]. Metric validations from researchers, e.g. utilized statistical nonparametric analysis [24], controlled study [25], and feasibility metric

level [26]. Realization in this study is for assessment purpose by completeness and empirical checking.

Completeness checking: This checking contains mapping between metrics, requirement statements, and threat statements as shown in Fig.3. For example, M1, M2, and M3 can drive the users to create strong patterns (RS1) that can slow down Misuser to find the correct pattern through Smudge Attack (TS1) or Guessing (TS3). Furthermore, the strong pattern also can makes difficult Misuser to find the correct pattern through Shoulder Surfing (TS2), Tricking (TS4), or Spyware (TS5). The completeness checking shows that all metrics have mapped to requirement statements and threat statements.

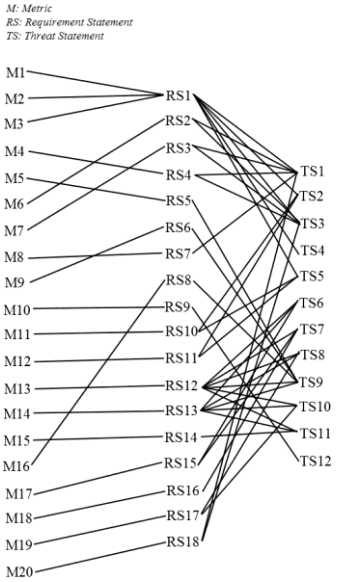

Fig.3. Mapping between Metric, Requirement Statement, and Threat Statement

Empirical Checking: The proposed security requirements metrics were evaluated against 30 Pattern-Lock applications on Android (labeled from Application I to XXX). Further explanation of empirical checking is given in Section IV.

IV. Results

This section provides activities for evaluating the metrics, data collection preparation, and data collection results.

A. Activities for Evaluating the Metrics

This stage contains four activities as follows:

- Data Collection. This activity can be mapped to Collection phase based on a measurement process from Roche (1995) in [15]. Data collected manually or automatically. Automatically if data displayed on the screen directly. Manually if data checked manually. Based on incident types and attributes from [23], this study uses location and symptom as attributes and fault as an incident type because this study evaluates the bugs on applications. The results of data collection are OK (Good) if there are intended features and capabilities, NG (Not Good) if there are intended features, but no capabilities, or NA (Not Applicable) if there are no intended features.
- Data Validation. Data validated based on good data characteristics from [23], i.e. correctness, accuracy, precision, consistency, time dependency, and replicability.
- Metrics Assessment. This activity can be mapped to Analysis and Interpretation phases based on a measurement process from Roche (1995) in [15]. This activity provides metrics assessment results. The results of metrics assessment are based on the interpretation of data collection result, i.e. SECURE, LESS SECURE, NA for OK, NG, NA respectively.
- Improvement Suggestion. This activity can be mapped to Feedback phase based on a measurement process from Roche (1995) in [15]. This activity provides suggestions to improve metrics achievement based on metrics assessment results.

B. Data Collection Preparation

The testing environment is a smartphone Samsung Galaxy Core GT-I8262 with Android version 4.1.2 JellyBean. The smartphone has been rooted before evaluating the applications using Kingo Android Root v2.0.5 [27]. The reasons are as follows. First, this study uses iKeyMonitor Android Free v4.7 [28] to validate the threat model that need rooting condition in order to capture screenshots. Second, to evaluate M14 (Ability to hide the locked application) also need rooting condition.

Technical considerations in this study are as follows. First, to evaluate M5 (Ability to lock remotely), the website *https://globfone.com/sms/* [29] is used as a substitution for smartphone to send locking or unlocking messages. Second, to evaluate M16 (Form of data storage), the "Show hidden files" in My Files-Menu-Settings has chosen to view hidden files and *android/data* has chosen as a location to find the data storage.

Nontechnical considerations in this study are as follows. First, this study only selects the App Locker types. The Screen Locker types did not evaluate because Android has provided the facility to lock the screen using Pattern-Lock, so that users tend to install the applications that have not provided on Android by default. Second, the selected applications must have more than 100.000 downloads when this study was conducted because more downloads mean the applications are popular, so that more users can take benefit from this study. Third, this study did not allow from same application developer to keep the objectivity of this research.

C. Data Collection Results

The results of data collection in this study are metrics assessment, security goals achievement, and metrics achievement.

1) Metrics Assessment Results

The metrics assessment result shows that most of 10

metrics are not applicable, i.e. M4 (2 of 30), M5 (1 of 30), M6 (14 of 30), M7 (6 of 30), M8 (11 of 30), M13 (9 of 30), M14 (2 of 30), M16 (1 of 30), M18 (8 of 30), and M20 (8 of 30). On the other hand, most of 10 metrics are applicable, i.e. M1 (30 of 30), M2 (30 of 30), M3 (30 of 30), M9 (19 of 30), M10 (30 of 30), M11 (20 of 30), M12 (19 of 30), M15 (30 of 30), M17 (20 of 30), M19 (30 of 30). The result is shown in Fig.4.

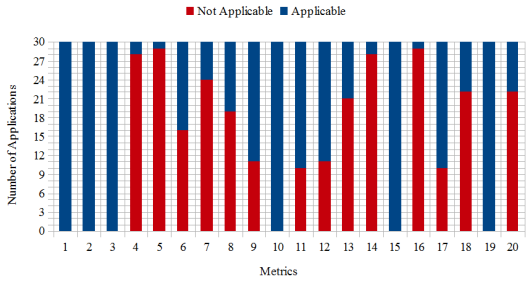

Fig.4. Metrics Assessment Result

This study utilized weighted values on metrics assessment results and obtains the values for metrics that can be classified in three categories as shown in Fig.5, i.e. high, medium, and low metrics. The high value metrics, i.e. M3: 1.00, M10: 0.97, and M1: 0.93. These three high value metrics will be many used to develop Pattern-Lock applications because from data collection found that many applications deal with the thresholds of proposed metrics. In other word, many applications have OK values for M3, M10, and M1. Therefore, Pattern-Lock application developers should consider M3, M10, and M1 first to build the Pattern-Lock applications. The medium value metrics, i.e. M2: 0.72, M11: 0.67, M17: 0,60, M19: 0.53, M9: 0.48, M6: 0.43, M12: 0.38, M8: 0.37, M13: 0.30, M20: 0.21, M18: 0.20, M7: 0.15, and M15: 0.10. Pattern-Lock application developers can consider the medium value metrics after satisfy the high ones. The low value metrics, i.e. M4: 0.07, M14: 0.05, M5: 0.03, and M16: 0.03. These four low metrics have low values because from data collection found that many applications not deal with the thresholds of proposed metrics. In other word, many applications have NG or NA values for M4, M14, M5, and M16.

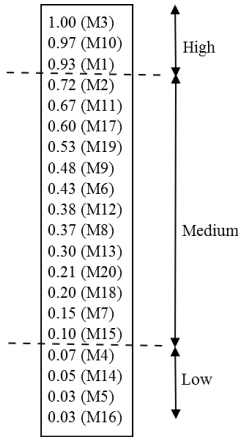

Fig.5. Categories Based on Weighting on Metrics Assessment Result

Both medium and low value metrics cannot consider as unimportant metrics because this classification is only based on data collection. This classification can be used as first insight to develop the Pattern-Lock applications. Next in discussion about metrics achievement results with weighted values will provide the classification based on metrics criticality in order to Pattern-Lock applications can act more as privacy protectors.

2) Security Goals Achievement Results

The metrics assessment results can be used to calculate the security goals achievements. The values range is from 0 (lowest) to 1 (highest). The achievements of specific security goals, i.e. G1 (Authorization), G2 (Confidentiality, Integrity, Availability), and G3 (Nonrepudiation) are 0.48, 0.45, and 0.21, whereas the global average value is (0.48+0.45+0.21) / 3 = 0.38. Pattern-Lock applications can meet the security goals if have values higher than the average of specific and global values. From 30 evaluated applications, only 5 applications that meet the security goals, both specific and global, i.e. Application XVII, XVIII, XXV, XXVI, and XXVIII. These results show that the security goals achievement for all evaluated applications is still low. Therefore, users should not have over high expectations when install the Pattern-Lock applications that the applications will fully locking the desired applications. The security goals achievement can be shown as a kiviat diagram in Fig. 6.

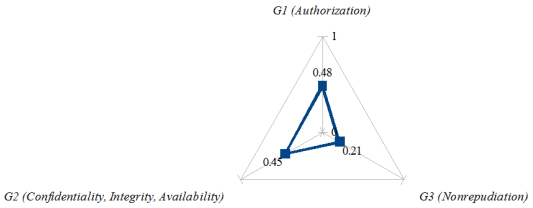

Fig.6. Kiviat Diagram for Security Goals Achievement

3) Metrics Achievement Results

In this study, metrics achievement results can be calculated without and with weighted values as follows.

Based on metrics achievement results without weighted values, the highest value is 13 (Application XXVI) and the lowest one is 4 (Application III, VI, VIII, XII, XIV, and XXVII). The average for all evaluated applications is still low, i.e. 7.47 (values range 0-20: lowest-highest). Therefore, Pattern-Lock applications should have the value higher than 7.47 according to the metrics achievement results without weighted values or equal with more than 7 metrics have achieved. From 30 evaluated applications, only 15 applications that have the value higher than 7. The ranking rules are number of OK, number of NG, number of downloads, and version number. The more OK, the better. If the number of OK is same, the number of NG will be considered. The more

NG, the better. If the number of NG is same, the number of NA will be same. Therefore, the number of downloads will be considered. The more downloads, the better because the application preferable by user. If the number of downloads are same, the version number will be considered. The higher version number, the better because the application more mature. The applications rank result without weighted values is shown as a column bar chart in Fig. 7.

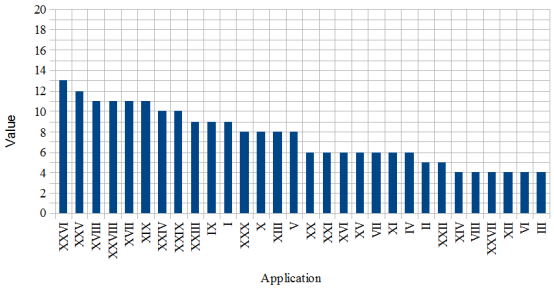

Fig.7. Column bar Chart for Applications Rank Result without Weighted Values

The metrics achievement without weighted values does not consider the weight of each metric. For example, M15 (Ability to keep locked consistently) should have higher value than M20 (Number of log information) because more critical. If M15 does not achieve, the content of locked applications will be open and reputation as a privacy protector will be decreased. Furthermore, if the Pattern-Lock application developers have limited resources, e.g. people, time, and energy, the developers will find difficult to achieve all proposed metrics. For that reason, this study has considered to prioritize the proposed metrics based on their weights as follows:

- The metrics that very critical for now are weighted by 3 (High), i.e. M15, M17, and M19, so that the total = 3x3 = 9. The reasons are as follows. First, if the lock is inconsistent, the content of locked application will be open for a while (M15). In addition, from the data collection results obtained that 27 of 30 applications are inconsistent. Second, as a privacy protector, before protecting other applications, the Pattern-Lock application shall be capable to protect itself, i.e. M17 and M19.
- The metrics that not enough critical for now are weighted by 2 (Medium), i.e. M6, M7, and M11, so that the total = 2x3 = 6. The failed login attempts limitation, artificial delay, and password hiding when entered by user are standard features for textual authentication that can be implemented on Pattern-Lock application, but many of them have not achieved the threshold yet, i.e. M6 (12 of 30 are Capable), M7 (3 of 30 are Increase), and M11 (20 of 30 are Capable).
- The metrics that least critical for now are weighted by 1 (Low), i.e. M1, M2, M3, M4, M5, M8, M9, M10, M12, M13, M14, M16, M18, and M20, so that the total = 1x14 = 14. The reasons are as follows. First, the features are standard for Pattern-Lock applications, i.e. grid size is 3x3 for M1, number of minimum dots are 4 for M2, number of pattern characteristics are 1 for M3, capable to relock for M10, and use non-plain text although data storage location is not found for M16. Second, the features have not widely implemented, i.e. M4 (2 of 30), M5 (1 of 30), M13 (9 of 30), and M14 (1 of 30). Third, only additional features to protect applications, i.e. M8. Fourth, without the features, the contents of locked application are still protected, i.e. M9, M12, M18, and M20.

Based on metrics achievement results with weighted values, the highest value is 0.759 (Application XXVI) and the lowest one is 0.138 (Application III, VI, VIII, and XXVII). The average for all evaluated applications is still low, i.e. 0.377 (values range 0-1: lowest-highest). Therefore, Pattern-Lock applications should have the value higher than 0.377 according to the metrics achievement results with weighted values. From 30 evaluated applications, only 14 applications that have the value higher than 0.377. The ranking rules are average value, number of NG, number of NA, number of downloads, and version number. First consideration is average value. The higher average value, the better. If the average value is same, the number of NG will be considered. The lower NG, the better. If the number of NG is same, the number of NA will be considered. The lower NA, the better. If the number of NA is same, the number of downloads will be considered. The more downloads, the better because the application preferable by user. If the number of downloads are same, the version number will be considered. The higher version number, the better because the application more mature. The applications rank result with weighted values is shown as a column bar chart in Fig. 8.

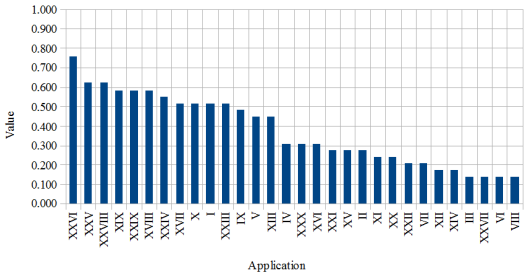

Fig.8. Column bar Chart for Applications Rank Result with Weighted Values

V. Discussion

This section discusses the proposed metrics, data collection results, and threats of validation.

A. Discussion for Proposed Metrics

The proposed metrics are especially defined for Pattern-Lock applications that have App Locker types.

However, most of them (18 of 20) also applicable for Screen Locker types. Only few of them (2 of 20) which not applicable for Screen Locker types, i.e. M4 (Ability to use different patterns to lock different applications) and M14 (Ability to hide the locked application).

The proposed metrics contain 15 nominal, 2 ordinal, 1 ratio, and 2 absolute scales. According to [23], closer to absolute is better. Furthermore, according to [5], [14], more quantitative is better. That is difficult. For example, to measure that applications can prevent from deleting (M17), the possible values are Capable, Not Capable, or Not Applicable and that is nominal scale and qualitative.

The proposed metrics do not have an Authentication goal because this study did not investigate the fake Pattern-Lock applications. The reasons are as follows. First, this study did not successfully find the fake Pattern-Lock applications when develops the threat model, so that the element S from STRIDE has not identified yet. Second, the users can report inappropriate applications that violate Google Play Store policy. Therefore, the Authentication goal is out of the scope of this study and open for future work.

The threshold and interpretation of the proposed metrics can be adjusted with the Pattern-Lock development in the future. For example, the threshold for M3 (Number of pattern characteristics that successfully detected) is 1 still SECURE. This is because pattern strength meter like in [6] is not widely implemented like password strength meter in textual authentication. If the password strength meter has widely implemented, the threshold for M3 will become more than 1.

Compared with [7] that have evaluated shoulder surfing, "pattern length" can compared with M1 and M2 because higher grid size and more dots will trigger longer pattern, "line visibility" can compared with M11, and "knight moves", "intersecting points", and "overlapping points" can compared with M3. So, the proposed metrics are not also considering the security aspects of pattern against shoulder surfing, but also pattern recovery (M12). Furthermore, compared with [3], [6] about pattern strength, "pattern length" can compared with M1 and M2, "direction changes", "entropy", "start and end points", "sub-pattern analysis", "ratio of non-repeated segments", and "number of intersecting points" can compared with M3. So, the proposed metrics are capable enough to measure pattern strength.

The data collection method still many checked manually that have a high probability to get erroneous data. This is a weak point for this study because a human has lower durability than a computer program that can check automatically. So, the automatically checking can be considered as complementing with manually one. For metric validation, the proposed metrics only utilize completeness and empirical checking. Maybe feasibility checking that proposed by [26] can be implemented in the future.

B. Discussion for Data Collection Results

Below are discussion for results of metrics assessment, security goals achievement, and metrics achievement.

1) Discussion for Metrics Assessment Results

The proposed metrics that have NA (Not Applicable) there are thirteen metrics, i.e. M4, M5, M6, M7, M8, M9, M11, M12, M13, M14, M16, M17, and M18. Another seven metrics (M1, M2, M3, M10, M15, M19, and M20) are applicable. However, the results may be OK or NG. The NA conditions can happen if Pattern-Lock application developers still considered the privacy as not so important aspect and more prioritized the usability than the security.

M3, M10, and M1 metrics will be many used to develop the Pattern-Lock applications because have the highest values. Therefore, Pattern-Lock application developers should consider M3, M10, and M1 first to build the Pattern-Lock applications. On the other hand, M4, M14, M5, and M16 have the lowest values and can be the last consideration for Pattern-Lock application developers. However, these four lowest metrics can not be considered useless or not applicable. These four metrics have the lowest values because from the data collection found that the Pattern-Lock applications that evaluated still have NG or NA. For other thirteen metrics (M2, M11, M17, M19, M9, M6, M12, M8, M13, M20, M18, M7, and M15) have medium values and can be considered after the three highest metrics have considered.

Many applications do not provide intended features. For example, feature to hide the pattern (M11). This study prefers NA to NG for that case because the applications are not developed to hide the pattern. Furthermore, if intended features only include in PREMIUM or PRO version, the metrics assessment results will be NA. For example, Application VII is free, but provides a PRO version that has more features, i.e. hide the Pattern-Lock application (M13). Only one application (Application XXX) can be estimated using non-plain text (M16). The form of data storage from other applications cannot estimated although folder locations are known. This is due to only developers that know exactly the location and form of data, including the pattern, and the proposed metrics are defined from the user's viewpoint. So, it is also NA. However, although the applications do not provide relock feature (M10), if capable to relock, this study will prefers OK to NA. For NG case, Application XXII has a feature to prevent uninstalling (M17), but Application XXII still can be uninstalled.

For now, Pattern-Lock application developers should have more attention for metrics that have high criticality, i.e. M15 (Ability to keep locking consistently), M17 (Ability to prevent the Pattern-Lock application deleting), and M19 (Ability to resist from interruption) because these metrics have more impact to privacy protector reputation. For example, only a few (3 of 30) that can lock consistently (M15). However, metrics criticality can be adjusted with the security and privacy need in the future.

The validation of metrics can be theoretically or empirically. This study has validated the proposed metrics empirically for assessment purpose from 30 Pattern-Lock applications on Android. This number has

chosen because statistically, a sample was considered big must be at least 30. In addition, Pattern-Lock applications on Android are heterogeneous because the applications have varying capabilities, so that this number can be considered as a representative of the Pattern-Lock applications population on Android. Although the proposed metrics evaluated on Android, but the proposed metrics can also used for Pattern-Lock applications on mobile devices with operating systems other than Android.

2) Discussion for Security Goals Achievement Results

The security goal achievement of G1 (Authorization) from all evaluated applications is still far from 1, i.e. 0.48. This is reflected that the Pattern-Lock applications that evaluated have no ensure the unauthorized access and pattern recovery misusing. Moreover, the evaluated applications also have no ensure to drive the users to create the strong patterns although M3 is good, but M1 and M2 are still not good.

The security goal achievement of G2 (Confidentiality, Integrity, Availability) from all evaluated applications is also still far from 1, i.e. 0.45. This is reflected that the Pattern-Lock applications that evaluated have no ensure the confidentiality of pattern recovery when entered, the confidentiality of Pattern-Lock application's location, the confidentiality of locked application's location, the confidentiality of content because lock inconsistency, to prevent application and log deleting, and to prevent crashing on Pattern-Lock application. However, this study has no enough information to determine the data storage because the test environment is not set to factory reset mode and only considered *android/data* for the data storage location. On the other hand, the Pattern-Lock applications have enough to ensure the confidentiality of pattern when entered.

The security goal achievement of G3 (Nonrepudiation) from all evaluated applications is also still far from 1, i.e. 0.21. This is reflected that log still has no considered as important thing yet because from 30 applications, 22 applications have no log and 4 applications have log that contain log information below the threshold of M20.

In overall, the security goals achievement from all evaluated applications is still low, i.e. 0.38. This is reflected that the Pattern-Lock applications that evaluated are still need to many improvements on security aspects, from the lowest to the highest are Nonrepudiation (G3), Confidentiality, Integrity, Availability (G2), and Authorization (G1).

3) Discussion for Metrics Achievement Results

The metrics achievement results, both without and with weighted values, are reflected that the Pattern-Lock applications that evaluated are still far from user's expectation. If compared the results between without and with weighted values, not many applications that have significance change instead of Application IV (up: 7), VII (down: 4), VIII (down: 4), X (up: 4), XX (down: 6), and XXX (down: 4).

The more an application meets the metrics, the closer to user's expectations as a privacy protector. To know user's expectations, this study has considered user's reviews on Google Play Store when identify the threat model. The five highest rank with weighted values, i.e. Application XXVI, XXV, XXVIII, XIX, and XXIX have >10.000.000, >5.000.000, >50.000.000, >1.000.000, and >100.000.000 downloads respectively and v6.6.6, v7.2.2, v3.1.2, v2.3.0.007, and v2.15.3 respectively. Therefore, Application XXVI, XXV, XXVIII, XIX, and XXIX are closer to privacy protectors. On the other hand, the five lowest rank with weighted values, i.e. Application XIV, III, XXVII, VI, and VIII have >500.000, >100.000, >10.000.000, >100.000, and >100.000 downloads respectively and v1.0, v1.0.2, v1.3.86, v1.0, and v1.0.1 respectively. Therefore, Application XIV, III, XXVII, VI, and VIII are still far from privacy protectors.

The application rank can be divided into three zones, i.e. upper zone (rank 1-10), middle zone (rank 11-20), and lower zone (rank 21-30). Based on these zones, the applications that have more downloads (>1.000.000) are 8 of 10, 3 of 10, and 3 of 10 respectively, and the applications that have higher version numbers (> v1) are 7 of 10, 4 of 10, and 1 of 10 respectively. So, in general, the applications that have more downloads and higher version numbers will have higher ranks. However, Application XXVII has more downloads (>10.000.000), but has lower rank. Unfortunately, this study does not investigate why the users like to install certain applications.

Based on the results of metrics assessment without and with weighted values, each only 15 and 14 applications that meet the metrics assessment. The difference of one application is Application XXX that not meet the metrics assessment with weighted values because it only has achieved 0.310 (< 0.377).

C. Threats of Validation

The results of this study may be not valid if the results meet one of the following conditions:

- The results of metrics assessment and security goals achievement may be not valid if Pattern-Lock application developers have more prioritized the security than the usability. Therefore, metrics that have low usability based on combination from Table 6 and Fig.3, i.e. M1, M2, M3, M4, M5, M6, M7, M8, M9, M11, M12, M13, M14, and M18, will not be problems. For example, Fig.5 showed that M4 has value 0.07. This value can be not valid if many developers have implemented the different patterns to lock the different applications. Moreover, Application III does not meet the specific and global security goals because still have NA values. This result can be not valid if the developer of Application III has implemented the features that have OK and NG values.
- The results of metrics achievement may be not valid because the calculation is not affected by the number of downloads. The number of downloads

are not the technical factor that determines the quality of applications, in this case are Pattern-Lock applications, but only like or dislike factor. Users can download the Pattern-Lock applications because the applications are free, have simple user interfaces, have small sizes, have no annoying ads, or users are not looking for the privacy aspect, but other aspects, for example performance aspect on Application XXVIII and XXX which also developed to boost the smartphone performance.

VI. Conclusions

This study has identified 12 threat statements, specified 18 requirement statements, and defined 3 goals, 7 questions, 20 metrics for Pattern-Lock applications on mobile devices. Pattern-Lock application developers should prioritize the three highest value metrics, i.e. Number of Pattern Characteristics that Successfully Detected, Ability to Relock, and Grid Size. Moreover, the developers should ensure that the Pattern-Lock applications which developed should have values above the average of security goals achievements, both specifics and global, and above the average of metrics achievements, both without and with weighted values. Although the proposed metrics evaluated on Android, the proposed metrics can also be used for Pattern-Lock applications on mobile devices with operating systems other than Android.

There are several aspects to explore as future work, such as add Authentication goal if spoofing threats have identified, adjust the thresholds, interpretations, and metrics criticality as the Pattern-Lock development in the future, make data collection more automatic without ignoring the manually checking, make metric scale type closer to absolute and more quantitative if possible, or validate the proposed metrics to more Pattern-Lock applications on mobile devices. In addition, it is necessary to optimize the metric development process, utilize factory reset mode for test environment to achieve same starting point for evaluating the applications, implement the metric feasibility checking, create a self-assessment from the Test Procedure of proposed metrics, investigate why the users like to install certain applications, display the results as scores in Google Play Store like User Review scores, or get feedback from industry by sending the results to each evaluated application developers or the proposed metrics to Pattern-Lock application developers.

References

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge Attacks on Smartphone Touch Screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, Berkeley, CA, USA, 2010, pp. 1–7.

[2] H. Gao, W. Jia, F. Ye, and L. Ma, "A survey on the use of graphical passwords in security," *Journal of Software*, vol. 8, no. 7, Jul. 2013.

[3] P. Andriotis, T. Tryfonas, and G. Oikonomou, "Complexity Metrics and User Strength Perceptions of the Pattern-Lock Graphical Authentication Method," presented at the International Conference on Human Aspects of Information Security, Privacy, and Trust, 2014, pp. 115–126.

[4] C. Kern, A. Kesavan, and N. Daswani, *Foundations of Security: What Every Programmer Needs to Know*, 2007th ed. Berkeley, CA : New York: Apress, 2007.

[5] R. Savola, "Information security evaluation based on requirements, metrics and evidence information," in *Proceedings of the 6th Annual Security Conference*, Las Vegas, NV, 2007.

[6] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh, "On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 2343–2352.

[7] E. von Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann, "Easy to Draw, but Hard to Trace?: On the Observability of Grid-based (Un)Lock Patterns," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 2339–2342.

[8] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, New York, NY, USA, 2013, pp. 161–172.

[9] V. R. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Paradigm," *Encyclopedia of Software Engineering 2 Volume Set*, vol. 1. John Wiley & Sons, pp. 528–532, 1994.

[10] E. von Zezschwitz, A. Koslow, A. De Luca, and H. Hussmann, "Making Graphic-based Authentication Secure Against Smudge Attacks," in *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, New York, NY, USA, 2013, pp. 277–286.

[11] W. Hu, X. Wu, and G. Wei, "The Security Analysis of Graphical Passwords," in *Proceedings of the International Conference on Communications and Intelligence Information Security (ICCIIS)*, 2010, pp. 200–203.

[12] A. de L. Simao, F. Sicoli, L. de Melo, and R. de Sousa, "Acquisition of digital evidence in android smartphones," *Australian Digital Forensics Conference*, pp. 116–124, Jan. 2011.

[13] Y.-C. Tsai and C.-H. Yang, *Physical Forensic Acquisition and Pattern Unlock on Android Smart Phones*. Springer Netherlands, 2013.

[14] I. Sommerville, *Software Engineering*, 9th ed. USA: Pearson Education, 2011.

[15] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. USA: McGraw-Hill Education, 2010.

[16] A. Shostack, *Wiley: Threat Modeling: Designing for Security*. USA: John Wiley & Sons, 2014.

[17] L. Røstad, "An extended misuse case notation: Including vulnerabilities and the insider threat," in *Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2006.

[18] G. Sindre and A. L. Opdahl, "Eliciting Security Requirements with Misuse Cases," *Journal Requirements Engineering*, vol. 10, no. 1, pp. 34–44, Jan. 2005.

[19] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on requirements engineering for information security*

(SREIS), 2005, vol. 2005, pp. 1–8.
[20] D. Firesmith, "Engineering Security Requirements.," *The Journal of Object Technology*, vol. 2, no. 1, p. 53, 2003.
[21] S. Islam and P. Falcarin, "Measuring security requirements for software security," in *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, 2011, pp. 70–75.
[22] A. A. Abdulrazeg, N. M. Norwawi, and N. Basir, "Security measurement based on GQM to improve application security during requirements stage," *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 1, no. 3, pp. 211–220, 2012.
[23] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous & Practical Approach*, 2nd ed. Pws Pub Co, 1996.
[24] N. F. Schneidewind, "Methodology for validating software metrics," *IEEE Transactions on Software Engineering*, vol. 18, no. 5, pp. 410–422, May 1992.
[25] V. R. Basili, L. C. Briand, and W. L. Melo, "A validation of object-oriented design metrics as quality indicators," *IEEE Transactions on Software Engineering*, vol. 22, no. 10, pp. 751–761, 1996.
[26] R. Savola, "On the feasibility of utilizing security metrics in software-intensive systems," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 10, no. 1, pp. 230–239, 2010.
[27] "The Kingo Android Root Website," 2015. [Online]. Available: https://www.kingoapp.com/.
[28] "The iKeyMonitor Key Logger Spy App Website," 2015. [Online]. Available: https://ikeymonitor.com/.
[29] "The Globfone Website," 2016. [Online]. Available: https://globfone.com/.

Authors' Profiles

Irfan Afifullah His research interest is in the field of information security-related measurement and evaluation.

Bayu Hendradjaya received a PhD Degree in Software Engineering from La Trobe University. He is currently a lecturer and a head of software Engineering Research Lab in Institut Teknologi Bandung (Indonesia). His research interests are in the area of software security, software process improvement, software measurements, and software V&V.

How to cite this paper: Irfan Afifullah, Bayu Hendradjaya,"Security Requirements Metrics for Pattern-Lock Applications on Mobile Devices", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.11, pp.1-13, 2016.DOI: 10.5815/ijcnis.2016.11.01

