ATTESTLENS: A LARGE-SCALE MEASUREMENT OF PLAY INTEGRITY ADOPTION IN ANDROID APPS

Collin MacDonald, PhD Candidate
Stephen Herwig, Assistant Professor
William & Mary, Williamsburg, Virginia

Abstract

Google's Play Integrity framework has replaced SafetyNet as the primary mechanism for app, device, and account attestation on Android, yet its real-world adoption remains poorly understood. We present AttestLens, a large-scale measurement of attestation usage across 125,421 APKs collected from AndroZoo in April 2025. Our dataset spans six app categories—popular, banking, cryptocurrency, government, gaming, and random—and includes a longitudinal set of 814 government applications.

We develop robust static markers to detect SafetyNet and Play Integrity, even under common obfuscation techniques, and use them to quantify adoption by category and by app characteristics such as downloads, release date, and rating. Our findings are concerning: roughly 90% of apps in every category reference neither framework, and adoption remains particularly low in security-sensitive areas such as banking, cryptocurrency, and gaming. Additionally, many government apps continue to reference SafetyNet despite its deprecation. Across 6,281 Play Integrity-invoking applications, we find that most attestation originates from third-party packages rather than first-party invocation. On a positive note, most invoking apps follow Google's recommendation of pairing Play Integrity with other defense-in-depth techniques.

1. Introduction

Attestation frameworks play a central role in Android security by gating access to sensitive features such as financial transactions, account recovery, and anti-abuse controls. Google's Play Integrity API [1] has replaced the deprecated SafetyNet Attestation API [2] and is now the recommended mechanism for verifying app, device, and user integrity before processing high-value actions. In public developer communications, Google has suggested that apps adopting Play Integrity features have an 80% reduction in unauthorized usage [3], underscoring the framework's intended role in strengthening anti-abuse protections.

In principle, Play Integrity offers stronger guarantees than its predecessor. It ties app integrity to Play-distributed packages, leverages hardware-backed security signals where available, and exposes richer verdicts on device state, Google Play Protect, and potentially abusive behavior. Google has also fully shut down SafetyNet as of January 2025, making Play Integrity the only first-party option for Google Play apps. In practice, however, we do not know how widely Play Integrity is adopted or the nature of apps that choose to rely on it.

These questions matter for both security and platform design. If adoption is low in high-risk categories (e.g., banking, crypto, government, and games with strong cheating incentives), then the ecosystem may not be getting the benefits Google intended. Whereas an earlier work, SafetyNot [4], examined how apps misused the older SafetyNet API, our focus is on ecosystem-wide adoption and usage of Play Integrity—a framework Google explicitly designed to eliminate such misuse. SafetyNet's deprecation also raises a natural migration question: when a core security service disappears,

do developers update their apps, remove checks, or simply leave broken code in place?

We address these questions with AttestLens, a large-scale measurement study of attestation framework usage in Android apps. Using AndroZoo [5, 6], we construct a dataset of 125,421 unique APKs with Google Play metadata, spanning six categories: popular, banking, cryptocurrency, government, gaming, and a random sample of apps. We also collect all historical versions of 814 government apps, enabling a longitudinal view of migration behavior around the SafetyNet deprecation. We identify static markers for SafetyNet and Play Integrity that survive common obfuscation, and we validate them on hand-built sample apps and decompiled binaries.

Applying these markers, we first quantify adoption: whether an app references any attestation framework, and if so, which one(s). We then relate adoption to app characteristics such as download count, initial release date, and average star rating. We find that roughly 90% of apps in each category include neither SafetyNet nor Play Integrity, and that Play Integrity adoption is heavily skewed toward highly downloaded "popular" apps, with banking, crypto, gaming, and government apps lagging behind. Among government apps, SafetyNet usage persists well past Google's end of support, and only a small fraction migrate to Play Integrity.

Next, we examine usage: how Play Integrity is actually invoked in code. Decompiling 7,334 APKs that contain Play Integrity-related strings, we distinguish between first-party and third-party usage and identify common integration patterns. We find that a majority of Play Integrity references reside solely in third-party packages. Finally, we explore Play Integrity's role in defense-in-depth, measuring co-occurrence with other hardening techniques such as root detection, runtime integrity checks, and certificate pinning.

Contributions. We make the following contributions:

- **Large-scale adoption measurement.** We present the first in-depth analysis of Play Integrity and SafetyNet adoption across 125k Android APKs, covering six app categories plus a longitudinal corpus of government apps.
- **Adoption factors and migration behavior**. We quantify how Play Integrity usage correlates with downloads, release date, and rating; and we characterize real-world migration away from SafetyNet in government apps.
- **Identification of third-party usage.** We analyze how developers integrate Play Integrity in practice, and provide strong evidence that third-party libraries (e.g., Firebase, ReCaptcha) drive Play Integrity's adoption.

Abreviated Paper Contents. This paper has been formatted to match the requirements of VSGC. The sections on related work, defense-in-depth, security implications, and future work have been removed and can be found in our full paper later this year.

2. Background

2.1. SafetyNet

Google introduced the SafetyNet Attestation framework in 2017 to strengthen Android's anti-abuse and device integrity protections. SafetyNet enabled developers to verify that an app was running on a genuine, non-rooted Android device and whether the app had been tampered with, such as through repackaging. According to Google, its goal was to help servers "distinguish traffic coming from genuine, compatible Android devices from traffic coming from less-trusted sources" [2].

SafetyNet operated through a secure process separate from the app itself: the *Google Mo-*

bile Services (GMS) component performed integrity checks and sent the results to Google's servers, which returned a signed attestation for developers to verify on their backends. Correct implementation, however, proved difficult, as SafetyNet required secure server-side nonce generation and verification; apps that mistakenly performed these steps on the client were vulnerable to bypasses [4]. Google officially deprecated SafetyNet in June 2022 [7] and fully ceased support in January 2025 [8].

2.2. Play Integrity

Play Integrity [1], introduced in February 2022, builds on and strengthens SafetyNet by providing more comprehensive, developer-friendly, and tamper-resistant attestation. In addition to detecting rooted or modified devices, it verifies that user actions originate from an untampered app installed via Google Play on a certified device, and reports broader risk signals such as missing security updates, active overlay or screen-capture apps, disabled Play Protect, and anomalous request patterns [9]. Like SafetyNet, Play Integrity gathers integrity evidence on the device and submits it to Google's servers, which return a signed and encrypted token containing verdicts about the app, device, and user. By leveraging hardware-backed security signals and server-side verification, Play Integrity reduces opportunities for client-side tampering and misconfiguration while enabling app servers to evaluate client trustworthiness.

Play Integrity offers two API modes—*Classic* and *Standard*—that differ in both usability and security tradeoffs. The Classic API [10], available on Android 4.4 and later, performs a fresh integrity assessment for each request and uses a developer-generated nonce to bind the attestation to a specific action. While this can reduce the time-of-check-to-time-of-use (TOCTOU) window, it incurs higher latency and requires developers to manage replay protections correctly. The newer Standard API [11], introduced in July 2023, is designed for low-latency, high-frequency checks. It leverages on-device caching of integrity signals, replaces the nonce with a cryptographic *requestHash* that binds the token to a specific server request, and delegates replay protections to Google Play infrastructure, reducing integration complexity.

3. Methodology

3.1. Data Source

We collect mobile applications and associated metadata for AttestLens from the AndroZoo dataset [5, 6] maintained by the University of Luxembourg. We focus exclusively on Google Play apps because Play Integrity applies only to Play-distributed apps, and AndroZoo provides detailed Play Store metadata (e.g., star ratings, see §4.3) for these entries. Since not all APKs originate from Google Play and some Play-based records lack metadata, we first merge and normalize these sources. We include apps with uploads dating back to 2017 to capture SafetyNet's introduction and analyze migration patterns, (specifically in our longitudinal study of government apps.)

The resulting dataset contains verified APK hashes and Google Play metadata for over 13 million Android apps. Each APK hash corresponds to a unique `(package_id, version)` pair. When duplicates occur—such as for multiple language builds—we retain only the most recently updated APK hash for each pair.

3.2. App Categorization

Downloading all 13 million APKs would have required over 100 TB of storage, making a full download impractical. Instead, we filtered the normalized dataset into six manageable categories: *(1)* popular, *(2)* banking, *(3)* crypto, *(4)* government, *(5)* gaming, and *(6)* a random sample of apps. These categories capture a broad range of applications, particularly those handling sensitive or financial data. Table 1 summarizes the selection criteria and statistics; apps may appear in multiple categories. For longi-

Table 1: App categories used in AttestLens, along with associated metrics.

Category	Filter Criteria	App Version	Valid APKs	Invalid APKs
Popular	1,000,000+ Downloads	Latest	69,312	7
Banking	Package name contains `bank`	Latest	12,847	0
Crypto	Package name contains `crypto`	Latest	4,371	0
Government	Package name starts with `gov.`	Latest	814	0
Games (partial)	Published game	Latest	20,000	0
Random	N/A	Latest	20,000	3
Government (longitudinal)	Package name starts with `gov.`	All	4,271	0

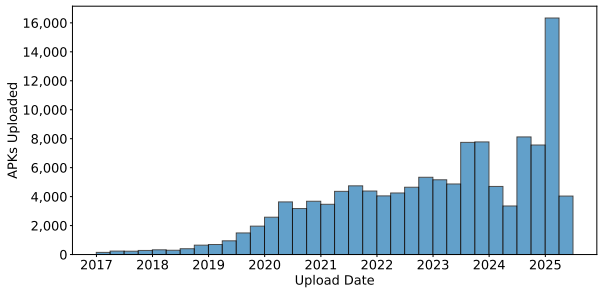


Figure 1: Histogram of APK uploads by date.

tudinal analysis, we download all available versions of each government-published app.

After filtering, we downloaded 125,421 unique APKs in April 2025, with only 10 failing verification;[1] 1,923 appeared in multiple categories. For string-based categories, we validated classification accuracy using random samples sized for approximately 95% confidence and ±5% margin of error. Our manual review yielded 100% accuracy for crypto (58 sampled apps) and government (56 apps), and 96.6% precision for banking (59 apps, 2 misclassifications). The APKs span uploads from 2017 to 2025 (Figure 1).

4. Adoption

Since its release in early 2022, the Play Integrity framework has been widely discussed in developer blogs and online forums [12, 13, 14, 15]. While past studies provided limited insights into Play Integrity adoption, none focused on the details of Play Integrity's global adoption [16]. In this section, we examine the adoption of Play Integrity across over 125,000 unique Android apps, including key factors that correlate with Play Integrity usage and broader adoption trends.

4.1. Detecting Framework Adoption

To detect the use of SafetyNet and Play Integrity in downloaded APKs, we first need to understand how each framework appears in compiled apps. We thus build a set of sample Java applications that use either SafetyNet, Play Integrity with the Standard API, Play Integrity with the Classic API, or no integrity framework at all. We compile each sample in both debug and release modes, and with and without code obfuscation via the R8 compiler [17]. Using the `strings` command-line tool, we extract and examine framework-related strings from each build to identify distinctive markers that persist even under obfuscation. We then use these markers to statically identify SafetyNet and Play Integrity usage in downloaded APKs without requiring decompilation. We call an app that has such a marker an *attestable app*.

4.2. Adoption by Category

Play Integrity adoption varies widely by application type, popularity, and publisher. Overall, most apps use neither SafetyNet nor Play Integrity. Among those that do, Figure 2 shows the proportion of attestable apps in each category, with Table 2 providing the corresponding counts. The following sections describe each category in detail.

[1] Each downloaded APK was hashed and compared to the hash provided by AndroZoo. If a downloaded APK's hash did not match the expected value, we attempted to redownload it up to 3 times before considering it `failed` and excluding it from future analysis.

Table 2: Adoption by category. (The attestation columns are mutually exclusive.)

Category	Total Apps	No Attestation	SafetyNet	PI			SafetyNet + PI (Either)
				Classic	Standard	Both	
Popular	69,312	60,100	2,788	1,800	92	4,005	527
Banking	12,847	11,521	1,005	143	6	98	74
Crypto	4,371	3,957	320	35	0	48	11
Government	814	710	88	6	1	6	3
Games	20,000	19,003	748	93	1	141	14
Random	20,000	18,170	1,433	175	1	88	133

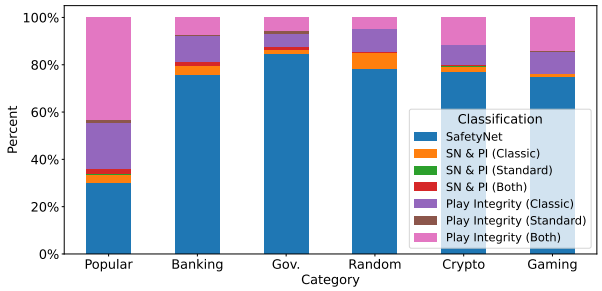


Figure 2: Percentage of attestable applications by category using an attestation framework.

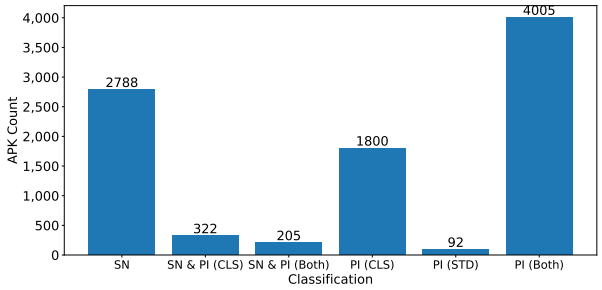


Figure 3: Play Integrity adoption among attestable popular apps.

Popular Apps. Popular apps—those with the highest download counts—show the strongest migration from SafetyNet to Play Integrity ($\chi^2(1, N = 13,883) = 2,719, p < 0.00001, V = 0.44$). In total, 13.3% of popular apps reference at least one attestation framework, similar to other categories. However, unlike other categories, most attestable popular apps adopt Play Integrity: 69.7% reference either the Classic or Standard API, the highest rate among all app groups. Figure 4 shows a detailed breakdown of framework usage for this category.

Banking apps. Despite being highly regulated and handling sensitive financial data, banking apps show limited use of Play Integrity: 89.7% contain no references to either SafetyNet or Play Integrity. Among the small subset that include attestation, 75.8% rely exclusively on the now-defunct SafetyNet. We infer that these apps either do not rely on SafetyNet to secure core functionality, or implement fail-open logic to prevent the shutdown of SafetyNet from affecting end users of the application.

Crypto apps. Crypto apps exhibit a nearly identical pattern: 90.5% do not use any attestation framework, and among those that do, 77.3% depend solely on SafetyNet. As with banking apps, Play Integrity adoption remains rare despite the security-sensitive nature of these services.

Government apps. Government-published apps show somewhat higher adoption: 12.8% reference at least one attestation framework. However, 84.6% of these rely only on SafetyNet, making this category the most SafetyNet-dependent of all.

Gaming apps. Gaming apps have the lowest adoption overall, with 95.1% containing no references to either SafetyNet or Play Integrity ($\chi^2(1, N = 127,344) = 855, p < 0.00001, V = 0.08$). This suggests that, despite Play Integrity support in Unity and Unreal Engine, most game developers are not using attestation for anti-cheating protection.

Random apps. Randomly sampled apps from Google Play showed similarly low adoption:

90.9% lack any reference to attestation frameworks, and among attestable apps, 78.3% use only SafetyNet. This is somewhat expected, as many apps—especially those without networked or server-backed functionality—have little need for attestation.

Adoption Insights by Category

- ~90% of applications do not include SafetyNet or Play Integrity.
- Among attestable applications, popular applications adopt Play Integrity at the highest rate (69.7%).
- Less than 5.0% of games published on the Google Play Store include SafetyNet or Play Integrity.

4.3. Adoption Factors

In this section, we examine how download count, release date, and average star rating relate to an app's adoption of SafetyNet or Play Integrity. We focus on the *random* category, noting that AndroZoo metadata includes these attributes for many—but not all—of our 20,000 *random* apps. Figure 4 summarizes our findings among the 1,830 attestable apps in this category.

Total downloads. Of the three factors, total downloads is the strongest predictor of Play Integrity adoption among attestable apps. As Figure 4(a) shows, 20% of apps with even 1,000 lifetime downloads include Play Integrity, and more than half of apps with at least one million downloads include Play Integrity (see Table 3). This trend is intuitive, as apps with more users are likely to generate more revenue, and therefore place a higher value in security features.

Release date. Google introduced Play Integrity in February 2022; we examine whether apps first published before and after this point include the framework at different rates. As Figure 4(b) and Table 4 show, apps released since 2023 include Play Integrity far more frequently than old apps ($\chi^2(1, N = 18,971) = 485, p < 0.00001, V = 0.16$) Surprisingly, older apps—those first published between 2012 and 2016—also show modest adoption. In contrast, apps released during SafetyNet's peak years of support (2017–2022) exhibit the lowest uptake.

Star rating. Play Integrity adoption shows little relation to an app's overall star rating (see Table 5; $\chi^2(4, N = 5,278) = 2.58, p = 0.63$). Although Figure 4(c) shows some variation between ratings, it is minimal for apps with an average rating above 1.5.[2] This result is not unexpected, as Play Integrity (and SafetyNet) are not inherently user-facing features.

Adoption Insights by Factor

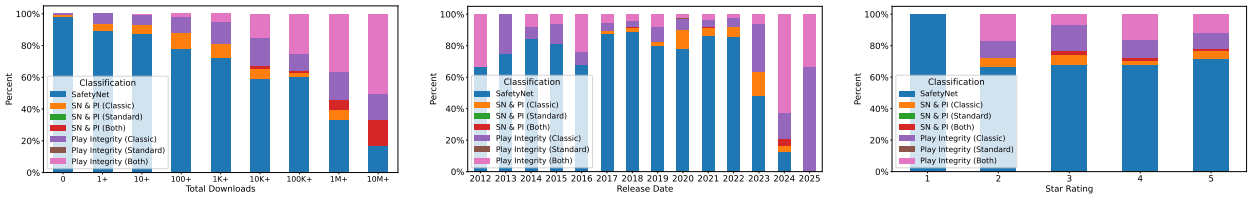
- Apps with high download counts and recent release dates show the strongest adoption of Play Integrity.
- An app's star rating has little relation to its adoption of Play Integrity.

4.4. Longitudinal Study: SafetyNet Migration in Government Apps

Google's retirement of the SafetyNet framework creates an opportunity to examine how developers respond to major service deprecations in mobile apps. Developers maintaining SafetyNet-based apps faced three choices: migrate to Play Integrity, remove SafetyNet entirely, or take no action. To understand how this transition played out in practice, we conduct a longitudinal analysis of attestation use in government-published apps.

We analyze more than 5,000 APKs spanning the 814 government applications and track which frameworks appear across different app versions. As Figure 5 shows, adoption gradually shifts from SafetyNet toward Play Integrity, but overall progress remains slow. By April 2025, only 15.4% of attestable government apps reference Play Integrity—well below the 21.7% observed in our random-app category, though only modestly supported by the data ($\chi^2(1, N = 1,934) = 2.33, p = 0.13, V = 0.03$). Alarmingly, we identify 90 apps that continue to refer-

[2] Star rating buckets were defined as 1 (1.0 to 1.5), 2 (1.5 to 2.5), 3 (2.5 to 3.5), 4 (3.5 to 4.5) and 5 (4.5 to 5.0).



(a) Adoption by Total Downloads (b) Adoption by Release Date (c) Adoption by Star Rating

Figure 4: Factors that influence adoption of Play Integrity among attestable *random* apps.

Table 3: Adoption by total downloads (*random* apps).

Total Downloads	Total Apps	No Attestation	SafetyNet	PI			SafetyNet + PI (Either)
				Classic	Standard	Both	
0	945	817	126	1	0	0	1
1+	2,864	2,630	210	15	0	0	9
10+	4,678	4,267	358	27	0	1	25
100+	4,786	4,352	338	42	0	9	45
1,000+	3,576	3,247	237	44	0	17	31
10,000+	2,000	1,825	104	30	1	26	14
100,000+	832	752	48	9	0	20	3
1,000,000+	255	222	11	6	0	12	4
10,000,000+	57	51	1	1	0	3	1
100,000,000+	7	7	0	0	0	0	0

Table 4: Adoption by release date (*random* apps).

Release Date	Total Apps	No Attestation	SafetyNet	PI			SafetyNet + PI (Either)
				Classic	Standard	Both	
2010	6	6	0	0	0	0	0
2011	34	34	0	0	0	0	0
2012	57	54	2	0	0	1	0
2013	125	117	6	2	0	0	0
2014	189	176	11	1	0	1	0
2015	340	324	13	2	0	1	0
2016	574	549	17	2	0	6	0
2017	1,015	940	66	4	0	4	1
2018	1,438	1,348	80	3	0	4	3
2019	2,747	2,584	130	15	0	13	5
2020	4,059	3,859	156	13	1	5	25
2021	3,766	3,393	321	18	0	12	22
2022	3,086	2,564	448	30	0	10	34
2023	1,370	1,130	116	72	0	15	37
2024	155	131	3	4	0	15	2
2025	10	7	0	2	0	1	0

Table 5: Adoption by star rating.

Star Rating	Total Apps	No Attestation	SafetyNet	PI			SafetyNet + PI (Either)
				Classic	Standard	Both	
1	37	34	3	0	0	0	0
2	174	156	12	2	0	3	1
3	724	681	29	7	0	3	4
4	2,681	2,474	140	23	1	33	10
5	2,112	1,938	125	18	0	20	11

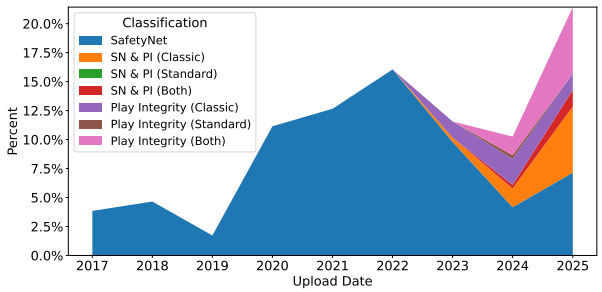


Figure 5: % of Government APKs uploaded by year with an attestation framework.

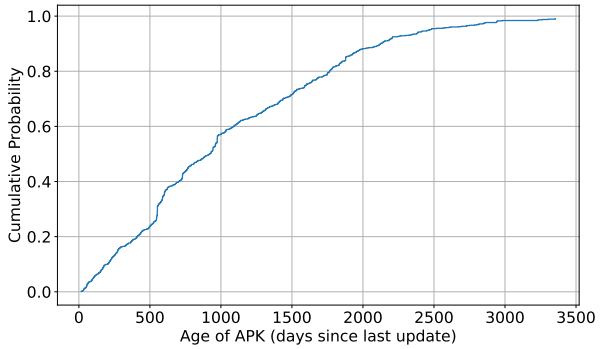


Figure 6: CDF of APK age for government apps.

ence SafetyNet without adopting Play Integrity, 7 that remove SafetyNet without adding any replacement, and only 9 that migrate from SafetyNet to Play Integrity. A potential cause for the lackluster reponse is that more than half of all government apps have not been updated in over two and a half years (see Figure 6).

SafetyNet Migration Insight

Few government apps have adopted Play Integrity, despite prior SafetyNet usage.

5. Usage

In this section, we move beyond detecting the mere presence of SafetyNet or Play Integrity and examine how Play Integrity is actually used, and when it is not.

5.1. Identifying Play Integrity Code

Our adoption analysis identified 7,334 apps containing Play Integrity strings. To study their usage, we decompile each APK with JADX [18], a tool that reconstructs Java source code from DEX and APK files. We decompile every app that we flagged as including Play Integrity in the first static-analysis phase (§4.1). For each decompiled APK, we scan all text-based files for an expanded set of Play Integrity markers, based on the Android Developer documentation [10, 11]. Using the package names, we distinguish between first-party and third-party usage. We call an app that contains Play Integrity markers outside of the Play Integrity library code itself an *invoking app*. Apps which include the Play Integrity library code but do not invoke it are *non-invoking apps*.

5.2. Code-level Results

After decompiling the 7,334 apps flagged as using Play Integrity, we successfully find Play Integrity library code in 7,323 of them—99.9% of the original set.[3] Table 6 summarizes high-level usage patterns. Only 341 apps use Play Integrity exclusively in first-party code. The majority (4,589 apps) combine first-party and third-party code, while 2,393 rely solely on third-party code. Figure 7 provides a detailed breakdown of Play Integrity code across all de-

[3]Six apps contain none of the usage-related markers, while five apps appear to invoke Play Integrity without including the Play Integrity library code; we exclude these eleven apps from further analysis.

Table 6: Where is Play Integrity code being used?

Metric	APKs
First-party	4,930
Third-party	6,982
Only First-party	341
Only Third-party	2,393
Only First-party & Third Party	4,589
Classic API	6,197
Standard API	1,492
Invoking Apps	6,281
Non-Invoking Apps	1,042

Table 7: Play Integrity in invoking applications.

API	First-party	Third-party	Both
Standard	22	9	53
Classic	111	1295	3383
Both	9	417	982

compiled apps by API type (Classic vs. Standard), first-party vs. third-party code, and Play Integrity library location.

Invoking Applications We identified invocation-related Play Integrity code in 6,281 APKs. Table 7 contains key metrics on Play Integrity usage for all invoking applications for both the Classic and Standard APIs. Usage is dominated by the Classic API and third-party code. Usage of the Standard API without the Classic API is exceedingly rare among both first and third party codebases, and represents less than 1.4% of all invoking applications.

Non-invoking Applications Despite a variety of obfuscation and compilation techniques present in the decompiled APKs, we successfully detected code for the Play Integrity library in 7,323 APKs, of which 1,042 were non-invoking apps. In these apps, we detected the presence of the Play Integrity library without detecting corresponding invocation code—suggesting an incomplete implementation, accidental library inclusion, or other development-centric mistake. Non-invoking apps include the Play Integrity library in first-party code (199), third-party code (672), and both first- and third-party code (171).

Third-party code. Through manual inspection, we identify 22 distinct third-party packages containing Play Integrity-related code across both invoking and non-invoking apps (Table 8). Google-published libraries account for most third-party usage, followed by analytics and advertising frameworks. Of these 22 libraries, 20 use the Classic API and only 7 use the Standard API. In many cases, third-party usage primarily protects the SDK provider's backend (e.g., CAPTCHA or authentication services), benefiting the host app only when those services gate core workflows.

Play Integrity Code-level Insights

- Among both first- and third-party code, usage of the Classic API is dominant.
- Over 30% of APKs only include Play Integrity through third-party code, making intentional adoption of Play Integrity less measurable.
- Google's Firebase Authentication package is present in over half of all APKs we analyzed.

6. Conclusion

In this work, we presented AttestLens, a large-scale study of Play Integrity adoption and usage across more than 125,000 Android APKs. Our findings show that adoption remains low across all application categories, including highly downloaded, highly rated, and security-sensitive apps. On a more positive note, adoption continues to increase year over year, and most developers who use Play Integrity use it as part of a broader defense-in-depth strategy.

References

[1] Google, LLC., "Play Integrity API Library release notes," Oct. 2025.

[2] Google, LLC., "SafetyNet attestation, a building block for anti-abuse," Apr. 2017.

[3] D. Elliot, "Making the Play Integrity API faster, more resilient, and more private," Dec. 2024. Ac-

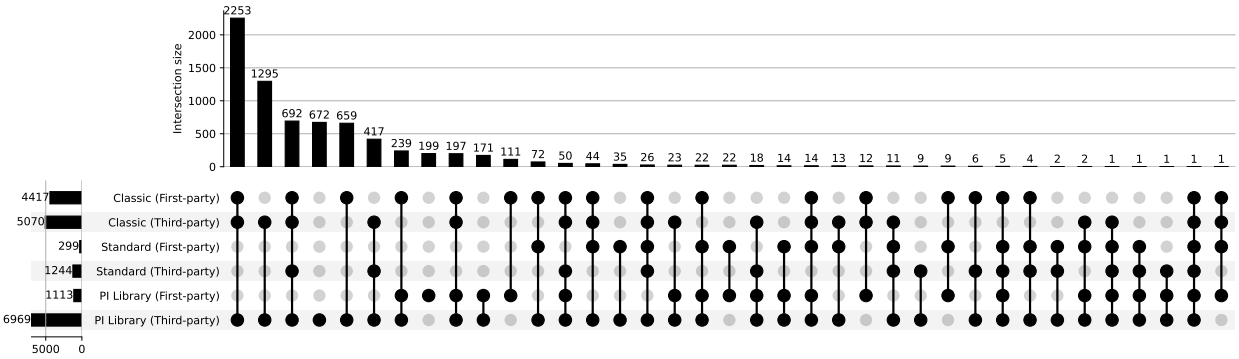


Figure 7: Play Integrity usage among invoking and non-invoking applications.

Table 8: Play Integrity findings in third-party code. Google-published packages appear in gray.

API Type	Package	APKs
Classic API	GMS	1,220
	ReCaptcha	2,373
	Firebase (App Check)	434
	Firebase (Auth.)	3,880
	Firebase (misc.)	840
	Google (misc.)	889
	AdJoe	88
	AppFlyer	510
	Daon	3
	FutureAE	6
	Games37	15
	GeoComply	5
	IdWall	5
	Protectt	13
	Radar.io	2
	RSA	2
	Shield Square	2
	Ticket Master	9
	VISA	14
	Yandex	71
Standard API	ReCaptcha	1187
	Firebase (misc.)	2
	Google (misc.)	3
	AdJoe	52
	GG Incent	3
	Radar.io	5
	Trusting Social	1

cessed: 2026-3-03.

[4] M. Ibrahim, A. Imran, and A. Bianchi, "SafetyNOT: On the usage of the SafetyNet attestation API in Android," in *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2021.

[5] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "Androzoo: Collecting millions of android apps for the research community," in *Proceedings of the 13th International Conference on Mining Software Repositories*, MSR '16, (New York, NY, USA), pp. 468–471, ACM, 2016.

[6] M. Alecci, P. J. R. Jiménez, K. Allix, T. F. Bissyandé, and J. Klein, "Androzoo: A retrospective with a glimpse into the future," in *Proceedings of the 21st International Conference on Mining Software Repositories*, pp. 389–393, 2024.

[7] SafetyNet API Clients Team, "Discontinuing the SafetyNet Attestation API," June 2022.

[8] Google, LLC., "About the SafetyNet Attestation API deprecation | Security," May 2025.

[9] Google, LLC., "Overview of the Play Integrity API," Oct. 2025.

[10] Google, LLC., "Make a classic API request," Oct. 2025.

[11] Google, LLC., "Make a standard API request," July 2025.

[12] Google, LLC., "Stronger threat detection, simpler integration: Protect your growth with the Play Integrity API," Nov. 2025.

[13] ElyeProj, "Play Integrity API, any potential issue of turning it ON?," Sept. 2024.

[14] XDA, "[GUIDE] ??? How to Pass Strong Integrity on Android (Step-by-Step Guide)," Apr. 2025.

[15] emaayan, "[HELP] is there a way to pass Strong Device Integrity by the play store?," Apr. 2024.

[16] M. Steinböck, J. Troost, W. Van Beijnum, J. Seredynski, H. Bos, M. Lindorfer, and A. Continella, "SoK: Hardening Techniques in the Mobile Ecosystem — Are We There Yet?," in *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*, pp. 789–806, June 2025. ISSN: 2995-1356.

[17] Google, LLC., "Enable app optimization," Sept. 2025.

[18] skylot, "skylot/jadx," Nov. 2025. original-date: 2013-03-18T17:08:21Z.