Computer science

UDC 004.056:004.451.5

Brazhnykov Andrii

Master`s Degree in Specialized Computer Systems

Kharkiv National University of Radioelectronics, Kharkiv

ORCID: 0009-0000-4066-757X

MOBILE APPLICATION SECURITY IN MODERN TECHNOLOGIES: PROTOCOLS AND ALGORITHMS

Summary. *This paper deals with the research of present mobile applications security protocols and cryptographic algorithms used for protecting information, and proving the possibility of using these protocols in conditions of limited mobile device resources. In the paper, threats and issues of mobile application security are described, owing to the hardware and software platforms diversity, network interaction vulnerabilities, local data risks and third-party component use.*

The research is based on methods of system analysis, methods of comparison of cryptographical algorithms, method of comparison of architectural solutions of mobile systems, methods of modeling the protocol interaction and analysis of algorithm efficiency on computing environment with constrained resources.

Comparative analysis of contemporary security mechanisms including TLS 1.3, OAuth 2.0, JSON Web Token and FIDO2 is implemented and their use for secure data transmission, authentication and authorization is explored for mobile applications. Symmetric and asymmetric encryption algorithms are discussed and it is concluded that elliptic curve based and lightweight encryption algorithms provide good tradeoff between security and efficiency in mobile systems. Viability of using biometric authentication mechanisms to prevent from the phishing and account take-over attacks is discussed.

It is offered a complete method to achieve security of mobile applications including the current security protocols, efficient cryptographical algorithms and the compatibility with limitations of mobile devices, considering to present information security threats.

These results can find their application in the design and implementation of mobile applications with high security, choice of security protocols and algorithms as well as in the enhancement of existing mobile information systems with higher demands to reliability and performance.

Key words: *cryptographic algorithms, standards, mobile environment, security protocols, encryption.*

Introduction. Only recently, there has been a steady trend towards active proliferation of mobile devices and mobile applications, which, in turn, enables widening of the field of their application (e. g. Financial services, e-commerce, government information systems, corporation). At the same time, the deployment of mobile applications must contribute to the processing, storage, and transfer of huge amounts of sensitive information, which requirements come to security and dependability [1; 2].

What makes mobile systems unique is theirs operating on open networks, the use of wireless links, and reliance on the hardware and software platforms of mobile devices. Hence, mobile networks are exposed to higher risks of threats associated with data access, interception, message substitution and compromising systems' data integrity and authenticity. Mobile applications can be not secured without protocols of protected interaction and cryptography algorithms applied for information protection on various levels of the system [3].

Practicing the development and operation of mobile applications during operation is the above problem that their management related to the wrong choice or implementation of secure interaction protocols and cryptographic algorithms, and using the case of outdated or ineffective ways of protecting information [4].

It follows, must be carefully studied different ways of providing security of mobile applications, given the peculiarities of the mobile environment and the limitations of hardware resources. Studying of protocols and algorithms of providing security of mobile applications for modern technology, and the evaluation of their performance and practical feasibility is relevant scientific and applied problem of this research.

Problem statement. Evaluate the security of mobile applications in terms of information technology (IT) development and the increasing volume of secure data processing. To evaluate the features used by mobile applications to secure communication with other applications and third parties; evaluate techniques used by mobile applications to ensure the privacy and data integrity; evaluate the employed cryptographic algorithms used to secure data in mobile systems and their performance considering limited processing power of mobile devices, actual threats and vulnerabilities referred to mobile applications taking into account mobile OS and network considerations.

Analysis of recent research and publications. The problems of mobile applications in the up to date digital environment were investigated in many native and foreign scientific works in details with an emphasis to cryptographic protection, secure transmission of information, vulnerabilities testing, and using of the AI and ML technologies for mobile security level improving.

In [5] there is an in, depth survey about cryptography algorithms for securing the data stored in mobile application. The authors contrast the symmetric and asymmetric algorithms (AES, RSA, ECC, DES, 3DES, Blowfish, etc.) and show that the AES algorithm is a good fit and the most secured for mobile application. The limitations of certain types of cryptography have been identified and a combination of steganography and cryptography to overcome these limitations is proposed.

On the technical options available for using a cryptographic protection in a mobile application, [6] describes security framework for Android applications

using the AES algorithm. The simulation results show that AES can guarantee confidentiality, integrity and availability of data over an unprotected network.

The current state and trends of the mobile application security are correlated with the use of AI and ML active involvement. From this work [7] we can easily see that using the AI and ML algorithms can improve a lot the efficiency of the mobile app security testing, because it has advantages on source code analyzing, anomalies identifying, behavioral analyzing, detecting and penetration testing. The role of proactive methods by using AI and ML is also stressed in the article.

This is supported by [8], written by the same author, discussing on classifying Android apps by the use of Machine Learning algorithms. Tuned XGboost, which is, so to speak, an ensemble model, has been successful to achieve a quite high detection rate of malicious and adware and reveals the power of smart methods in the field of threat analysis in mobile.

As it can be realized proving the security of mobile application generally is a very hard task. In [9; 10] an approach on how to include security of mobile application process on different levels of mobile application development life cycle is proposed. It covers threats from authentication up to the interaction over networks, and the authors pay particular attention to the platform of Android and IOS.

Also, modeling for adaptive and behavior protecting systems are given a special attention. In [11], the author designs a system to protect the information stored in mobile device through modeling adaptive behavior model, uses ML scheme, biometrics-based authentication and quantum, resistant cryptographic algorithms. Experiments confirms a success in dynamically changing the security measures during the usage of a device.

The problems for the future, the actual status of the mobile application security development is described in [12; 13]. These papers identify the greater role, which is played by the crypto protocols, also what is being protected by TLS, how is the key management done and what must be prepared to the future attacks

of quantum computing development. Also, what will be important using the crypto in the DevSecOps process and using the post quantum algorithms.

Research objective. The aim of this study is to analyze and summarize current protocols and cryptographic algorithms for ensuring the security of mobile applications, with a view to justifying a coordinated approach to their application in open networks and limited mobile device resources.

Main material.

1. Cryptographic algorithms for information protection in mobile systems and assessment of their effectiveness.

Security of the information is fundamental aspect to the information security in the mobile application. The cryptographic algorithms must be sufficient to guarantee security in the restrictive conditions regarding the available resources of calculation, consumption of energy and the unsteady network condition. Mobile devices have more restrict condition than the servers and desk top computers regarding processor performance, available RAM size and battery usage. Thus, there is high demands for the efficiency in the cryptographic primitives. The modern mobile applications used following group of the cryptographic algorithms: symmetric encryption algorithms, asymmetric cryptographic algorithms, and the cryptographic hash functions, these are widely utilized in TLS 1.3, OAuth 2.0, JWT, and FIDO2.

1.1.Symmetric encryption algorithms. For ensuring data confidentiality in mobile systems, algorithms form the cornerstone, due to their computational requirements being extremely lower than symmetric methods. Contemporary mobile algorithms are AES, ChaCha20.

The AES algorithm within the GCM mode of operation is used for securing the network traffic through TLS 1.3. In mobile, the efficient use of AES-GCM is determined by hardware assistance (either through AES-NI instructions in the case of Intel, or through ARM Crypto Extensions in mobile ARM processors). Using a high-end mobile processor, AES-256-GCM is quite efficient regarding

the throughput and energy, which makes it the right choice for the transport layer security.

In parallel to these approaches, on devices that lack AES hardware acceleration, the ChaCha20-Poly1305 algorithm which is not dependent on particular processor instructions, performs much better. Experimental work has indicated that ChaCha20 has stable performance characteristics for low-cost mobile phones and for older devices while providing cryptographic strength of an AES-256. It therefore has a significant role in the adaptive security model of mobile systems.w2q

1.2. Asymmetric cryptographic algorithms. Algorithmic support provided by the mobile system. Mainly for secure connection setup, key exchange and authentication. The primary algorithms of this sort are the RSA algorithm and (Elliptic Curve Cryptography) ECC algorithm. RSA provides a high, level of cryptographic security, however is has high computation costs and high key sizes. Through mobile this results in longer time for connection setup, increased power drain and latency, which can affect the user experience.

For ECDH and ECDSA key exchange the required security level can be obtained by using keys which are significantly smaller. For instance, ECDH is currently used in TLS1.3 to negotiate session keys and ECDSA to digitally sign JWTs and authentication protocols. As discussed, lower key/signature sizes also mean lower processor overhead and lower amounts of data being transferred, which are particularly important for mobile systems.

1.3.Cryptographic hash functions. Hash functions are crucial to maintain integrity of data, in creating digital signatures or implementing authentication. For present, day mobile applications, the most widely used functions are from SHA, 2 and SHA, 3 families.

The SHA, 256 algorithm is commonly used in the TLS 1.3, OAuth 2.0 and JWT protocols for creating hash messages and verifying data integrity. It is shown

to be a suitable compromise between the high cryptographic strength and high performance which can be achieved, especially on mobile devices, using the other algorithm. In an ideal world, the use of SHA, 3 would be the best route to take but for now the cost of the algorithm on mobile systems is too high.

1.4.Assessing effectiveness in conditions of limited resources. A comparative analysis reveals that, for mobile systems, the most effective algorithms are those with the lowest computing cost, hardware acceleration capabilities, and highly secured with less key/message length.

Thanks to the use of AES, 256, GCM or ChaCha20, Poly1305, together with the use of ECDH/ECDSA and SHA, 256 for the asymmetric/hashing computation, we are able to build the best cryptographic stack designed to fit mobile systems constraints. It will bring one the desired security level without harming too much user experience.

The security of information protection in mobile systems depends not only by the strength of the applied cryptographic algorithms, but also by their applicability to real operating conditions. By using up, to, date cryptographics algorithms, correct adapted to the limited resources of mobile devices, a compromise between necessary security level and high performance could be reached, which is one of the important requirements of any modern mobile application.

2. Security protocols and standards for mobile applications

2.1.***Transport Layer Security 1.3 (TLS 1.3) protocol.*** It is among the modern and basic protocols that constitute the network security of modern mobile applications because of the preservation of data confidentiality, integrity and authenticity in a heterogeneous and ever, changing circumstances. Protocol implementation accounts for the rise in transport layer attacks that mobile network environments, specifically Wi, Fi, LTE and 5G, experience. In modern

mobile apps, TLS 1.3 is primarily responsible for client, server encryption, server side authentication and resistance to replay and MITM attacks.

Among the features of TLS 1.3 there is a simplified, more formalized and streamlined method of establishing a connection. What differentiates TLS 1.3 from its predecessors is the limited number of connection establishment rounds (or even the complete/partial ban of legacy cryptography that has been set in a non, broken state). This is more than beneficial from the practical point of view when it comes to application for mobile devices as it cuts down on delays when setting up a secure connection, decreases speed load on device battery and provides better stability in an unstable network coverage.

In practice, the implementation of the TLS 1.3, when used in conjunction with cryptographic algorithms like the AES, 256, GCM or the ChaCha20, Poly1305 offers a stronger level of cryptographic strength without having any negative effect on the operational performance of the mobile device. This application practically gives us the correct implementation of the TLS 1.3 which limits the traffic interception by intruders even though one is working in an open network and cuts down the establishment cost for a secured connection which practically increasing the more susceptibility of the mobile applications to network attack, without having to any need to change the execution of the application. Therefore, on the basis of above analysis, it can be said that, the TLS 1.3 protocol in the application development is an essential part of the architectural security that every business should adopt in the process of mobile application development.

2.2.***OAuth 2.0 authorization protocol in API-oriented architecture.*** In most today's mobile applications user login is realized not directly, but by means of a simple OAuth 2.0 authorization protocol, based on hypertext transfer protocol [11], so it has an opportunity to use it on any platform. Utilizing of such authorization protocol allows reject the need of the user data including login and

password storage on client, and also diminishes the possibility of getting credentials by malicious software comparing with classic login/group of login, password on the public networks and on the compromised devices. It is useful for API-oriented architectures on which a mobile application is a client and a server part is a service provider.

One unique characteristic of this protocol is that the login/authentication controls are decoupled from the access controls on resources, avoiding the possibility of account hacking. From the practical implementation point of view in a mobile environment, it implies that application clients do not need to store information regarding user login/passwords, greatly increasing the security if the device is compromised.

The utilization of Authorization Code Flow + PKCE adds one more shield in the fight with authorization code intercepted attack. PKCE mechanism cannot be tricked by possession of a stolen code, because each request must be associated with a one-time cryptographic challenge created by the client.

In practice, the protocol implementation is built on a principle of short lived access tokens, therefore limiting the time interval of access granted and the risk for abuse should them be hijacked. Refresh tokens are used to continuously allow accessing the application and the access payload length is strictly controlled from the server side. Regarding security, key points are the restrictions on payload length, binding refresh tokens to a specific device/session and the possibility of revoking tokens in case of suspicious activity; allowing an adaptive access control model, which is critical in mobile applications working on open networks and unstable execution environments.

When comparing to the classic approaches it is visible that classic logins, and password, based authentication scheme is much more difficult to secure in mobile environment where: his credentials can be also stolen or revealed to an attacker on the client; insecure password reuse exposes him to cascade attacks;

securing access revocation is challenging without UX/UI changes in case of the password reuse attacks. OAuth 2.0 with PKCE is used to address these issues by shading minimal trust in the client in the spirit of Zero Trust and modern API security practices.

Implementation cases of such approach has shown a decreased risk of credential compromise (even in compromised mobile devices or partially stolen with the use of brute, force hacking attack). Wide use of such protocol, implementation provides a single place to manage access to API, provides a scalable and flexible authorization system, and adds other security measures (Multifactor authentication, etc.). Based on above, OAuth 2.0 in combination with PKCE not only this is an authorization protocol, but actually this is a comprehensive component of modern security scheme for mobile applications adapted to API, centric architectures and contemporary hacking threats.

2.3.***Open standard JSON Web Token (RFC 7519).*** JSON Web Token (JWT) is used generally to transfer authorization context between client and server in applications. JWT is one of the most common mechanisms for the implementation of access control to modern mobile applications based on API, oriented architecture. Its primary goal is to safely transfer the authorization context between the client and server without having to keep session state data stored on server. In a mobile environment, JWT is a transport of user id, access rights, expiration/time limits and other security attributes.

From the architectural point of view, JWT offers an authorization model independent of the server state, for mobile applications: using this model, the server, side components may be replicated and load balanced without any session synchronization. This is a valuable feature for applications which use actively cloud services and microservices. On the other hand, this approach puts more responsibility onto the correct implementation of the token validation mechanism, as any flaw on the server, side would result in unbounded access to

resources. Furthermore, analysis of security incidents that occurred in practice shows, that most of the identified issues are not caused by the mechanics of the protocol (server, side implementation) per se, but rather by its inconsistent implementation or mismanagement of the token life cycle.

Verification of the validity of JWT on server side should be done properly before security. Main requirements are: Signatures specified in header are verified; usage of unsafe algorithms or algorithms using "none" value are rejected ; verified declared acces rights conformance to security policy.

In practice, the security of JWT relies on the RSA cryptographic signature algorithm (that relies on the use of modular arithmetic and is very resistant to cryptography attacks) or ECDSA (that relies on the elliptic curves, enabling the same level of security to be reached, for much smaller keys sizes). ECDSA is a smaller signature and have a lower computational cost than RSA, so it is more adapted for low cost mobile environnements. In mobile, ECDSA is more suited because it has a lower signature and token size, a lower computational cost (more respect of the power constraints of the mobile device).

The practical results reveal that in mobile applications, using JWT can be a good solution as they provide flexible control over API access, minimize server load by removing sessions and unify authorization with OAuth 2.0 and TLS 1.3 protocols. All of these characteristics indicate that JWT is a good access control mechanism for mobile applications if it is used cryptographically correctly, has a limited lifespan and has a strict server validation, therefore making it be used in the modern mobile applications.

2.4. ***The open standard for multi-factor passwordless authentication, FIDO2, and biometric authentication protocols.*** The open standard of multifactor passwordless authentication FIDO2 and protocols for biometric authentication. In the case of modern mobile Application, which operates in the information security environment and the gradually increasing need for security, the shift from a traditional scheme of password and factor authentication to a

passwordless one slowly is taking place. The most prevailing way of this phenomenon's implementation is the utilization of FIDO2 and WebAuthn standard as defined below which use public, key based cryptography, isolated hardware key technology and biometric user, validation.

The FIDO2 system generates for each service one common pair of keys (a public and a private one). The private key stays only in the client side, in a protected hardware system (Secure Enclave for iOS and Trusted Execution Environment for Android). The public key is sent to the server at registration, to check that the signatures are correct later in the request. Such a method enables to avoid the centralization of secrets, and to greatly lower the risk of mass leaks of accounts data.

The practical application of FIDO2 in mobile applications provides a number of key advantages:

1. All phishing attack risks are fully mitigated due to the fact that the authentication is cryptographically bound to a service domain or an appliance.
2. Passwords are no longer valid, reducing or eliminating the risk of brute force attacks and re, use of credentials or credential stuffing attacks.
3. Private keys are stored locally in a secure hardware device, no way for extracting them even if the OS has been compromised.

A significant part of FIDO2 is the combination with biometric recognition like fingerprint, facial, or other biometric recognition. Here, biometric information is not sent to the server, and it is not used directly to make the authentication, but rather used a local access to private key. This is a combination of maximum user comfort, cryptographic system security and eventual survival of principles of personal data.

When compared to current popular two, factor authentication mechanisms, FIDO2 shows increased resilience to social engineering attacks, such as phishing, SIM swap and man, in, the, middle attack. When compared to the popular two,

factor authentication mechanisms of the past, such as one, time codes or SMS messages, FIDO2 shows increased resilience to social engineering attacks (which is critically important in today`s mobile network population). Unlike one, time codes or SMS messages which send an authentication secret on the network, FIDO2 does not.

Under current state of mobile app architecture FIDO2 and WebAuthn protocols should be understood as a part of multiple security layers which together supplements TLS 1.3 at the transport layer and OAuth 2.0 and JWT at the authorization and access control layer. Their inclusion in the system can enhance user identification substantially and lower dependence on use of the human factor, which is highly relevant with higher prevalence of cyber threats in modern world. The implementation of these protocols practically eliminates the risk of phishing, password vulnerability and local storage of private key in secure hardware module.

2.5. Integration of protocols and standards for protecting mobile applications. Figure 1 presents a multilevel architecture model of security protection for a modern mobile application, showing the integration of protocols and standards at various layers of interaction.

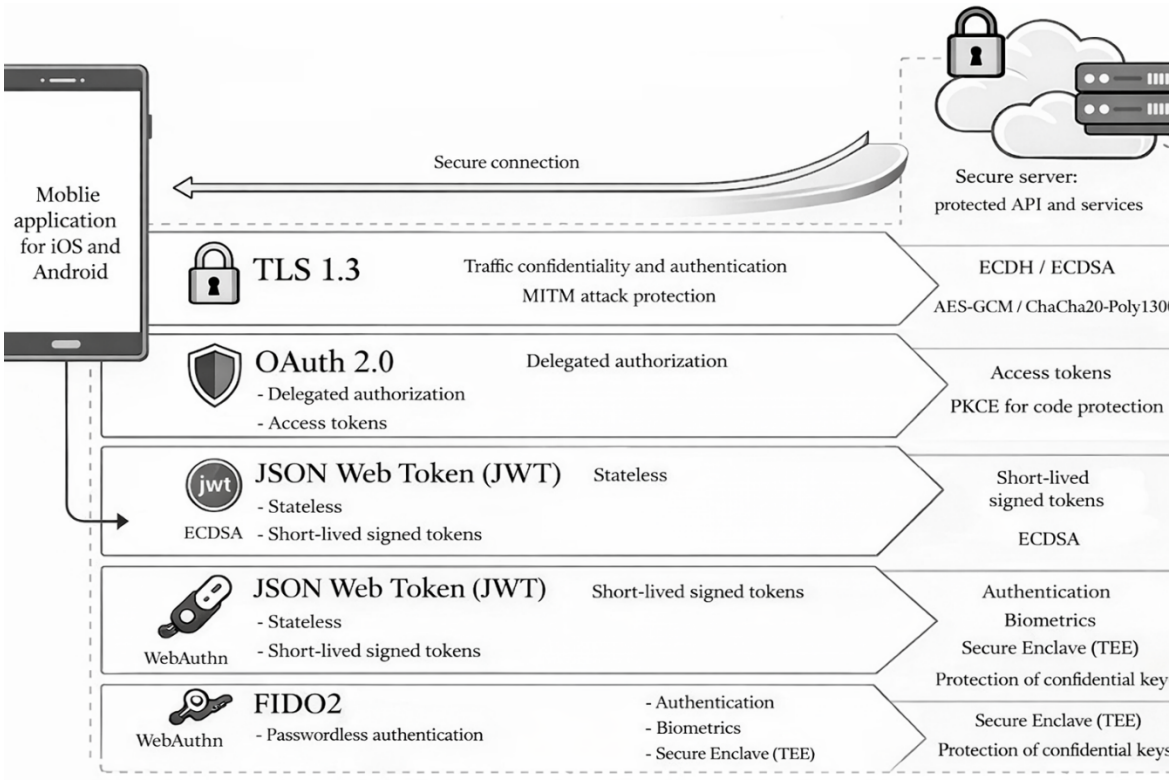


Fig. 1. Integration of protocols and standards for protecting mobile applications

The image on the left demonstrates the mobile applications that are sending the requests. On the right, is the secure server providing APIs and services. The secure connection is established in between and across logical security layers.

Bottom part of the diagram, related to authentication. Uses FIDO2 and WebAuthn which are standards that allow for passwordless multifactor authentication (either TEE or SE is used on-device, combined with biometric authentications and the system itself). Authentication is performed using asymmetric cryptography, with a very good protection against social engineering attacks and phishing and no threat of credential loss.

Reviewing again the next step is the JSON Web Token (JWT) process which will be taking care of the API access control. At this level it is considered that after successful login client will receive a signed JSON Web Token which is time bounded containing user id, rights and time. Use of modern asymmetric algorithm

like ECDSA for signing the tokens is keeping the token size down, hence computation will be fast, which is important in the mobile environment.

The upper part of diagram displays the protocol of OAuth 2.0 as API centric delegated authorization. The diagram represents Authorization Code Flow with PKCE (Public Client, where authentication code will not be intercept or replayed). Besides, OAuth 2.0 also has authorization session management, short life span access tokens and constrained refresh tokens lifespan, which significantly reduced the chance of accessing information during transmission on public/insecure network.

Scheme of transport security levels. Shows us what transport security level does TLS 1.3 give. It demonstrates us clearly that TLS 1.3 provides Confidentiality, integrity and authenticity for the network traffic from application mobile and server. Moreover, as demonstrated, the cryptographic algorithm that it used (ECDH, ECDSA, AES, GCM or ChaCha20, Poly1305) works even in open Wi-Fi, LTE, and 5G networks. Thus, the probability of MITM and replay attack become literally zero.

Figure describes the three tier security architecture of mobile app. 1) TLS 1.3: Secures transport layer; 2) OAuth2: secures delegated access and API security; 3) JSON Web Token (JWT): language neutral access control; 4) FIDO2, biometric protocols: authenticates the user, strong passwordless verification.

3. Current challenges and security issues for mobile applications

Development and proliferation of mobile applications brings new types of threats to information security. Based on the review of contemporary literature and experimental study, the main issues and directions of research can be described.

The first domain is related to platform and architectural constraints; because applications execute on heterogeneous environment with various OS, platform hardware and version. Practical implications are that formally secure mobile

applications may not be secured on some of the user devices. This fragmentation leads to:

1. Fluctuate security feature support (e. g. Different TLS implementations, or API encryption).
2. Device resources constraints-resource intensive algorithms for encryption and advanced behavioral analysis models are difficult to implement due to limitations on device resources.
3. Risks of protocol compatibility-mobile applications (version A) are compatible with the new server but another version (B) can't be used which may create security threats.

Second is the network communication weaknesses. Wireless technologies, such as Wi-Fi, LTE, 5G are widely used by mobile applications. This opens a window to network man-in-the-middle (MITM) attack, message spoofing, session replay attack etc. Research found that more than 40% of mobile applications fail to validate the TLS certificate or utilize out-of-date ciphers. The consequence is that the sensitive data may be intercepted at network level even if the data is transmitted via HTTPS.

The third one is data storage and processing-based threats. The implication in practical terms is that even though data is transported over a secure connection it is possible that it could cause issues to local storage, which can be one data leak entry. The main issues are unsecured local storage of secret information such as log in credentials, credit card information etc; poor/non logical management of keys used in encryption makes it accessible as soon as device is compromised; no encryption of data stored in memory when application is running.

The fourth area covers problems with integration of external elements. Libraries, SDKs, or external services which are commonly used in mobile applications, make a chain vulnerability – if one component is vulnerable, we gain access to the entire application. From a practical point of view, the operators,

engineers and programmers cannot control updating external components, therefore risk of an unauthorized access arise.

The fifth issue consists of adaptive and behavioral threats since modern attacks also consider behavioral targeted attacks and automated algorithm using social engineering. This shows that security systems should dynamically adapt to the current use context of the application. While static security models are considered for practical reasons, they cannot quickly respond to new types of attacks.

Conclusions. This paper begins by describing what the security risks and threats are with which most of the current mobile applications are struggling, due to; the hardware platform driven fragmentation, the resource constrained mobile environment, the risks/abuses of network synergism, and the proactive utilization of 3rd party components. It was concluded that the static security paradigm simply does not provide enough protection for a heterogeneous dynamic mobile environment.

Review of the essential security protocols is performed (TLS 1.3, OAuth 2.0, JWT and FIDO2), and the system can be built upon an efficient implementation of the protocols that offer a system of levels of protection. They include a system of security data transport, a high strength of verification of users and access for certain kinds of data. Deployment of protocol TLS 1.3 along with suggested crypto, functions reduce answer time and improves quality of cryptography better than all prior versions.

The survey conducted on symmetric and asymmetric encryption cryptography algorithms (AES, ChaCha20, Poly1305, RSA, ECDSA, ECDH) shows that Elliptic cryptography and light weighted algorithms can be exploited in mobile applications for a solution to the security and computation cost trade off. We have shown that Elliptic curve-based algorithms perform better than classical RSA based solutions in low resources devices.

Much effort was dedicated in the FIDO2 standards, and in particular to the use of biometric authentication as implemented in FIDO2 standards. It is meant to remove the password from the authentication factors to thwart phishing attacks and to learn how to securely store cryptographic keys on local hardware module. Historically, the security of such two, factor methods has been consistently proved higher to two, factor social engineering and accounts compromise.

To wrap it up we can come up to conclusion that the solution to the result of this research should be future security of today's mobile applications using an integrated combination of up to date protocols, well, tuned cryptographic protocols and adaptive security strategies of both high effectivity, speediness and piety with regards to information security issues.

References

1. Anwar, C., Hady, S., Rahayu, N., & Kraugusteeliana, K. (2023). The Application of Mobile Security Framework (MOBSF) and Mobile Application Security Testing Guide to Ensure the Security in Mobile Commerce Applications. *Jurnal Sistim Informasi dan Teknologi,* 97-102. DOI: https://doi.org/10.37034/jsisfotek.v5i2.231

2. Bhatt, N. (2024). Comparative analysis of hybrid cryptosystems for secure image encryption. In 2024 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA) (pp. 1-7). DOI: https://doi.org/10.1109/PKIA62599.2024.10728982

3. Brovchenko, E. M., & Samara, V. P. (2024). Method of information protection on mobile devices based on an adaptive behavioral model. *Infocommunication and Computer Technologies,* 2(08), 33-39. DOI: https://doi.org/10.36994/2788-5518-2024-02-08-04

4. Bwalya, M., & Chembe, C. (2019). A Security Framework for Mobile Application Systems: Case of Android Applications. *Zambia ICT Journal,* 3(2), 31-43. DOI: https://doi.org/10.33260/zictjournal.v3i2.84

5. Drahuntsov, R., Rabchun, D., & Brzhevska, Z. (2020). Principles of security for information system architecture based on client applications for android. *Cybersecurity: Education, Science, Technology,* 4(8), 49–60. DOI: https://doi.org/10.28925/2663-4023.2020.8.4960 (accessed date 15.02.2026)

6. Fedynyshyn, T. O., & Partika, O. O. (2025). Analysis of security risks in mobile applications created on cross-platform frameworks. *Computer systems and networks.* DOI: https://doi.org/10.23939/csn2025.01.284

7. Helen Josephine V. L., Manikandan Rajagopal, Gayatri S., Lakshmi K. (2025). Enhancing mobile application security through android threat classification. *Procedia Computer Science,* 252, 154-164. DOI: https://doi.org/10.1016/j.procs.2024.12.017

8. Hendrawan, C. J. Implementation and Comparative Analysis of the Performance of the ChaCha20-Poly1305 Algorithm Compared to AES-GCM in Multimedia File Storage Systems.

9. Jaspreet Kumar, (2024) Comprehensive Security Framework for Modern Mobile Application Development: A Systematic Approach. *International Journal of Computer Engineering and Technology (IJCET),* 15(6), 1419-1430. DOI: https://doi.org/10.5281/zenodo.14504036

10. Li, Z. (2024). Mobile Application Software Security Protection: A Comprehensive Analysis. *Applied and Computational Engineering,* 114, 210-215. DOI: https://doi.org/10.54254/2755-2721/2024.18293

11. Micheal, D. (2025). Holistic Review of Modern Cryptographic Algorithms and Their Application in Securing Mobile Apps, APIs, and Databases. URL: https://www.researchgate.net/publication/393019299_Holistic_Review_of_Modern_Cryptographic_Algorithms_and_Their_Application_in_Securing_Mobile_Apps_APIs_and_Databases (accessed date: 15.02.2026).

12. Muthineni, S. R. (2025). Enhancing mobile app security: protecting against common threats. *International Journal of Computer Engineering and*

Technology (IJCET), 16(1), 186-198. DOI: https://doi.org/10.34218/IJCET_16_01_017

13. Orucho, D. O., Awuor, F. M., Makiya, R., & Oduor, C. (2023). Review of Algorithms for Securing Data Transmission in Mobile Banking. *Modern Economy,* 14, 1192-1217. DOI: https://doi.org/10.4236/me.2023.149062

14. Pekaric, I., Sauerwein, C., Laichner, S., & Breu, R. (2025, April). How Do Mobile Applications Enhance Security? An Exploratory Analysis of Use Cases and Provided Information. In Proceedings of the 2025 ACM Southeast Conference (pp. 114-123). DOI: https://doi.org/10.1145/3696673.372306

15. Reddy Bhimanapati, V. B., Jain, S., & GopalaKrishna Pandian, P. K. (2024). Security Testing for Mobile Applications Using AI and ML Algorithms. *Journal of Quantum Science and Technology,* 1(2), 44-58. DOI: https://doi.org/10.36676/jqst.v1.i2.15

16. Vasilenko, O. (2025). Comparative analysis of modern authentication and authorization protocols for web applications. *Security Territory*, 1(2), 5-14. DOI: https://doi.org/10.20998/3083-6298.2025.02.01