

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



ĐỒ ÁN TỐT NGHIỆP

NGHIÊN CỨU KỸ THUẬT GIẤU TIN
SỬ DỤNG CÔNG NGHỆ VR, AR

Ngành: An toàn thông tin
Mã số: 7.48.02.02

Sinh viên thực hiện:

Trần Thanh Thiện

Lớp: AT12L

Người hướng dẫn:

TS. Phạm Duy Trung

Khoa An toàn thông tin – Học viện Kỹ thuật mật mã

TP. Hồ Chí Minh, 2020

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



ĐỒ ÁN TỐT NGHIỆP

**NGHIÊN CỨU KỸ THUẬT GIẤU TIN
SỬ DỤNG CÔNG NGHỆ VR, AR**

Ngành: An toàn thông tin
Mã số: 7.48.02.02

Sinh viên thực hiện:

Trần Thanh Thiện

Lớp: AT12L

Người hướng dẫn:

TS. Phạm Duy Trung

Khoa An toàn thông tin – Học viện Kỹ thuật mật mã

TP. Hồ Chí Minh, 2020

MỤC LỤC

MỤC LỤC	i
DANH MỤC KÍ HIỆU VÀ VIẾT TẮT	iii
DANH MỤC HÌNH VẼ	iv
LỜI CẢM ƠN.....	v
LỜI NÓI ĐẦU	vi
Chương 1. TỔNG QUAN VỀ THỰC TẾ ẢO VÀ VẤN ĐỀ GIẤU TIN TRONG THỰC TẾ ẢO.....	1
1.1. Khái niệm và phân loại.....	1
1.1.1. Video 360 độ	1
1.1.2. AR – Thực tế ảo tăng cường	1
1.1.3. VR – Thực tế ảo.....	1
1.1.4. MR – Thực tế ảo kết hợp	2
1.1.5. XR – Thực tế ảo mở rộng	2
1.2. Lịch sử hình thành và phát triển.....	3
1.3. Ứng dụng công nghệ thực tế ảo vào các lĩnh vực.....	4
1.4. Vấn đề an toàn trong sử dụng công nghệ thực tế ảo	5
1.5. Cơ bản về kỹ thuật giấu tin.....	6
1.6. Phân loại kỹ thuật giấu tin.....	7
1.6.1. Kỹ thuật giấu tin LSB.....	8
1.6.2. Kỹ thuật giấu tin Wu-Lee	9
1.7. Phương pháp giấu tin trong các đối tượng đồ họa	10
1.7.1. Phương pháp giấu tin trong thực thể ba chiều	10
1.7.2. Phương pháp giấu tin trong video.....	12
1.8. Khả năng ẩn giấu và tương tác thông tin sử dụng thực tế ảo.....	12
1.9. Kết luận chương 1.....	13
Chương 2. PHÂN TÍCH, THIẾT KẾ MÔI TRƯỜNG THỰC TẾ ẢO CHO ỨNG DỤNG.....	14
2.1. Giới thiệu yêu cầu ứng dụng.....	14
2.1.1. Yêu cầu của ứng dụng thực tế ảo tăng cường (AR)	14
2.1.2. Yêu cầu của ứng dụng thực tế ảo (VR).....	15
2.2. Phân tích thiết kế mô hình ứng dụng.....	17
2.3. Chuẩn bị kiến thức, khả năng tích hợp vào ứng dụng.....	23
2.3.1. ARCore và Sceneform.....	23
2.3.2. Định dạng glTF trong giấu tin trong với ứng dụng AR.....	26
2.4. Kết luận chương 2.....	34
Chương 3. TRIỂN KHAI ỨNG DỤNG MINH HỌA GIẤU TIN TRONG MÔI TRƯỜNG THỰC TẾ ẢO	35
3.1. Yêu cầu đối với ứng dụng	35

3.2. Phát biểu bài toán và xây dựng kịch bản	35
3.2.1. Phát biểu bài toán.....	35
3.2.2. Xây dựng kịch bản mô phỏng	37
3.3. Xây dựng ứng dụng minh họa.....	39
3.3.1. Giới thiệu ứng dụng.....	39
3.3.2. Xây dựng hệ thống lưu trữ tập tin nhị phân mật	40
3.3.3. Thu nhận thông điệp và ẩn giấu thông tin	42
3.3.4. Phân tích khả năng an toàn của thông điệp và phương hướng pháp triển.....	52
3.4. Kết luận chương 3.....	53
KẾT LUẬN.....	54
TÀI LIỆU THAM KHẢO	55
PHỤ LỤC	57

DANH MỤC KÍ HIỆU VÀ VIẾT TẮT

⊗	Phép XOR bit theo module 2
AR	Augmented Reality
HMD	Head mounted display
MR	Mixed Reality
LSB	Least Signification Bit
VR	Virtual Reality
XR	Extended Reality

DANH MỤC HÌNH VẼ

Hình 1.1. Máy mô phỏng thực tế ảo Sensorama (nguồn Google).....	3
Hình 1.2. Lược đồ chung cho quá trình giấu thông tin.....	6
Hình 1.3. Lược đồ chung cho quá trình giải mã thông tin	7
Hình 1.4. Lược đồ phân loại các phương pháp giấu tin.....	8
Hình 1.5. Thay đổi giá trị thứ yếu của điểm ảnh để che giấu chuỗi bit.....	9
Hình 1.6. Mô hình biểu diễn lưới tam giác.....	11
Hình 1.7. Phương pháp xác định tam giác kế tiếp dựa trên khóa nhị phân	11
Hình 1.8. Phương pháp xây dựng tam giác giấu tin và mô hình lưới ba chiều	13
Hình 2.1. Mô hình che giấu thông tin áp dụng công nghệ AR và mật mã.....	17
Hình 2.2. Mô hình mã hóa đối xứng và bất đối xứng kết hợp.....	19
Hình 2.3. Thuật toán One – Time – Pad (OTP)	19
Hình 2.4. Hệ thống máy chủ lưu trữ đối tượng ba chiều trong không gian thực....	21
Hình 2.5. Cấu trúc đường ống (pipeline) nội dung 3D trước đây	27
Hình 2.6. Cấu trúc đường ống nội dung 3D kết hợp glTF	28
Hình 2.7. Mô tả về cấu trúc của định dạng glTF.....	29
Hình 2.8. Áp dụng nhúng tin vào cấu trúc glTF.....	30
Hình 2.9. Cấu trúc cây JSON trong glTF	31
Hình 3.1. Giai đoạn phân tích định dạng và ẩn giấu thông điệp.....	36
Hình 3.2. Mô hình mật mã giấu tin áp dụng trong ứng dụng AR	37
Hình 3.3. Mô hình triển khai giấu tin cơ bản đơn đối tượng.....	38
Hình 3.4. Minh họa cơ sở dữ liệu thời gian thực trên database	40
Hình 3.5. Firebase Storage – kho lưu trữ tệp tin	41
Hình 3.6. Giao diện làm việc nền tảng của ứng dụng.....	43
Hình 3.7. Giao diện đăng nhập tài khoản	44
Hình 3.8. Quét và phát hiện bề mặt đặt vật thể 3D.....	44
Hình 3.9. Đặt vật thể 3D trong thế giới thật	45
Hình 3.10. Giấu tin vào đối tượng 3D và hiển thị trên bề mặt thật.....	45
Hình 3.11. Tài nguyên của mô hình 3D mô phỏng	46
Hình 3.12. Các thành phần cơ bản trong cấu trúc JSON – glTF.....	47
Hình 3.13. Bảng khái quát các phân đoạn đệm với accessor tương ứng	49
Hình 3.14. Nguyên tắc đọc ghi dữ liệu nhị phân trong bộ đệm đối tượng 3D.....	51
Hình 3.15. So sánh giữa giấu tin bit cao với giấu tin bit thấp	52

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án tốt nghiệp này, tôi đã nhận được sự giúp đỡ tận tình từ giảng viên hướng dẫn TS. Phạm Duy Trung – Khoa An toàn thông tin Học viện Kỹ thuật Mật mã. Nhờ những lời khuyên và kiến thức mà thầy truyền đạt cũng như sự trợ giúp từ các bạn trong cùng nhóm nghiên cứu, mà tôi có thể thực hiện đồ án này.

Xin cảm ơn tất cả mọi người đã tạo những điều kiện tốt nhất để tôi hoàn thành đồ án tốt nghiệp.

SINH VIÊN THỰC HIỆN ĐỒ ÁN

Trần Thanh Thiện

LỜI NÓI ĐẦU

Thực tế ảo (Virtual Reality - VR) và thực tế ảo tăng cường (Augmented Reality – AR) không còn là một khái niệm xa lạ đối với các nhà phát triển công nghệ. Ứng dụng kỹ thuật này chỉ vừa được biết đến rộng rãi trong những năm cuối 2017, đầu năm 2018 và phát triển mạnh ngay sau đó với nhiều thành tựu nổi bật trên đa lĩnh vực. Có thể kể đến như: Giáo dục, kiến trúc, kỹ thuật đo đạc, giải phẫu học, bảo tàng ảo thuật...

Dẫn đầu trong việc đầu tư và phát triển các nguồn lực thực tế ảo nổi tiếng thường kể đến là Google - không ngừng cung cấp các tài liệu và nguồn thư viện như Google VR hay AR Core để xây dựng môi trường thực tế ảo cho ứng dụng (Android mobile) nói riêng và các nền tảng nói chung. Ngoài ra còn các công ty phát triển khác chưa được đề cập như: Facebook, Samsung, Niantic,...

Nền công nghệ mới ra đời, đòi hỏi các công ty và doanh nghiệp có những cải tiến mới về công nghệ. Sự cạnh tranh trong việc đi đầu công nghệ khiến các công ty dẫn đến những tranh chấp và hành vi tiêu cực. Bên cạnh đó, việc che giấu thông tin là một trong những vấn đề bảo mật về kinh doanh hoặc có thể là một chiêu trò dẫn dụ của một tổ chức hay cá nhân nào đó nhằm đưa thông tin ra bên ngoài ứng dụng công cộng mà khó bị phát hiện.

Trên thực tế, công nghệ thực tế ảo khá mới mẻ so với môi trường làm việc tại Việt Nam. Hầu như có rất ít các trường đại học đưa công nghệ này vào giảng dạy như một môn học, đồng nghĩa với việc hiểu rõ về cơ chế và hoạt động là khá thấp. Chính vì việc này dẫn đến khả năng tội phạm công nghệ cao có thể sử dụng nền tảng này để thực hiện các hành vi phạm pháp như: Ẩn giấu các tệp tin độc hại; tuyên truyền văn hóa phẩm tiêu cực, phản động; làm lộ thông tin tổ chức, cá nhân,... (không kể đến việc sử dụng giấu tin cho mục đích tích cực).

Việc che giấu thông tin trong ứng dụng hay một vật mang tin cụ thể không quá xa lạ đối với các nhà phát triển đang làm việc trong lĩnh vực an toàn thông tin, nhưng với việc che giấu thông tin sử dụng công nghệ thực tế ảo và thực tế ảo tăng cường là một bước tiến mới. Đồng thời việc này có thể chỉ rõ ra rằng, tội phạm công nghệ cao có thể có những đột biến đáng ngờ về phương thức và hành vi phạm tội.

Chính vì lý do đó, đề tài “ **Nghiên cứu kỹ thuật giấu tin sử dụng công nghệ AR/VR**” được chọn để thực hiện đồ án tốt nghiệp nhằm tìm hiểu và xây dựng một môi trường thực tế ảo có khả năng chứa đựng thông tin bí mật trên nền tảng Android.

Từ đó, cho thấy rằng mức độ an toàn thông tin trong giám sát hoạt động của các công nghệ mới là một điều vô cùng cần thiết.

Mục tiêu đặt ra khi thực hiện đồ án là:

1. Hệ thống lại kiến thức về kỹ thuật giấu tin.
2. Tìm hiểu các kiến thức trong lĩnh vực thực tế ảo nói chung và phương pháp giấu tin trong thực tế ảo nói riêng.
3. Xây dựng mô hình triển khai giấu tin trong ứng dụng thực tế ảo để làm rõ các kiến thức đã nghiên cứu.
4. Viết chương trình mô phỏng giấu tin trong ứng dụng thực tế ảo dựa vào mô hình đã trình bày.

Sau khoảng thời gian thực hiện đồ án, các mục tiêu cơ bản đã thực hiện được. Tuy nhiên kỹ thuật giấu tin áp dụng công nghệ thực tế ảo còn nhiều rào cản phức tạp, thời gian thực hiện đồ án tương đối ngắn nên không tránh khỏi thiếu sót. Rất mong được sự góp ý của các thầy cô, cũng như các bạn học viên để đồ án này được hoàn thiện hơn.

SINH VIÊN THỰC HIỆN ĐỒ ÁN

Trần Thanh Thiện

CHƯƠNG 1. TỔNG QUAN VỀ THỰC TẾ ẢO VÀ VẤN ĐỀ GIẤU TIN TRONG THỰC TẾ ẢO

1.1. Khái niệm và phân loại

Công nghệ thực tế ảo (VR) hay thực tế ảo tăng cường (AR) không còn là những khái niệm xa lạ đối với những nhà phát triển công nghệ. Thuật ngữ thực tế ảo không phải là một khái niệm mới, tuy nhiên để nói về sự rộng rãi và mức độ phổ biến vẫn chưa được nhiều người trải nghiệm.

Công nghệ thực tế ảo (VR – virtual reality) là thuật ngữ miêu tả một môi trường mô phỏng bằng máy tính. Có khả năng tương tác hoặc không, với người dùng theo thời gian thực (real-time interactivity) nhờ vào các cảm biến (sensor). Người dùng có thể bị cuốn hút vào sự mô phỏng này nhờ vào tác động vào các giác quan như: Nhìn (thị giác), nghe (thính giác), sờ (xúc giác),... Các nhà nghiên cứu đang hi vọng có thể tạo ra các cảm giác khác như: Ngửi (khướu giác), nếm (vị giác)... [3].

Dựa vào các đặc tính hiển thị và khả năng tương tác giữa các người dùng với môi trường ảo người ta chia thực tế ảo thành 5 thể loại:

1.1.1. Video 360 độ

Đây là dạng sơ khai nhất của thực tế ảo, video được quay ở 360 độ, các kịch bản quay được xây dựng sẵn, người xem chỉ xem mà không thể tương tác với các cảnh quan xung quanh. Để xem được thể loại này cần trang bị một Cardboard hay HMD – một loại thiết bị cho phép người dùng xem nội dung thực tế ảo. Một số người không xem video 360 độ là thuần thực tế ảo hay VR [2][3].

1.1.2. AR – Thực tế ảo tăng cường

Khi người dùng sử dụng smartphone hoặc thiết bị ghi hình có camera; với ứng dụng hỗ trợ, công nghệ này sẽ xử lý hình ảnh để hiển thị vật thể 3D tương ứng với độ xa gần trong hoạt cảnh thật và những thay đổi vật lý; hoặc hiển thị trên bề mặt vật thể với xử lý 2D. Người dùng có thể tương tác với hiển thị này thông qua ứng dụng trên thiết bị thông minh hoặc cảm biến quan sát hành vi của người dùng và các thay đổi của đối tượng đó [3][4].

1.1.3. VR – Thực tế ảo

Công nghệ này có thể đưa người dùng trải nghiệm hòa mình vào một thế giới hoàn toàn khác biệt đối với thế giới thực tại thông qua lăng kính ảo. Khả năng nhận thức với thế giới thật xung quanh bị hạn chế do tác động của kính thực tế ảo mang lại. Khả năng tương tác với thế giới ảo tăng lên, nhiều cảm biến được trang bị để tối

ưu hóa trải nghiệm: Cảm biến đầu, cổ, tay, chân, mắt... nhằm theo dõi chuyển động người dùng để đưa ra xử lý đồ họa phù hợp [2][3].

1.1.4. MR – Thực tế ảo kết hợp

Hay còn gọi là thực tại lai (Hybrid Reality) là sự kết hợp giữa vật thể ảo và không gian thật nhằm tạo ra môi trường xóa mờ ranh giới thực và ảo. MR được phát triển bởi Microsoft qua kính Hololens [2][3][4].

1.1.5. XR – Thực tế ảo mở rộng

Công nghệ này áp dụng tất cả các công nghệ trên (trừ video 360 độ). Khả năng mà ranh giới giữa thực tại và thế giới ảo không còn rõ ràng nữa. Đây là kết quả mà công nghệ đang hướng tới trong tương lai [3].

Khác nhau cơ bản giữa các thể loại: VR mang người dùng vào thế giới ảo, trong khi MR và AR thì mang thế giới ảo vào thế giới hiện tại.

Nhìn chung, người dùng có khả năng tương tác với các sản phẩm đồ họa này mà không bị giới hạn về mặt không gian. Cảm nhận giữa thế giới thật và ảo là không hoàn toàn xác định.

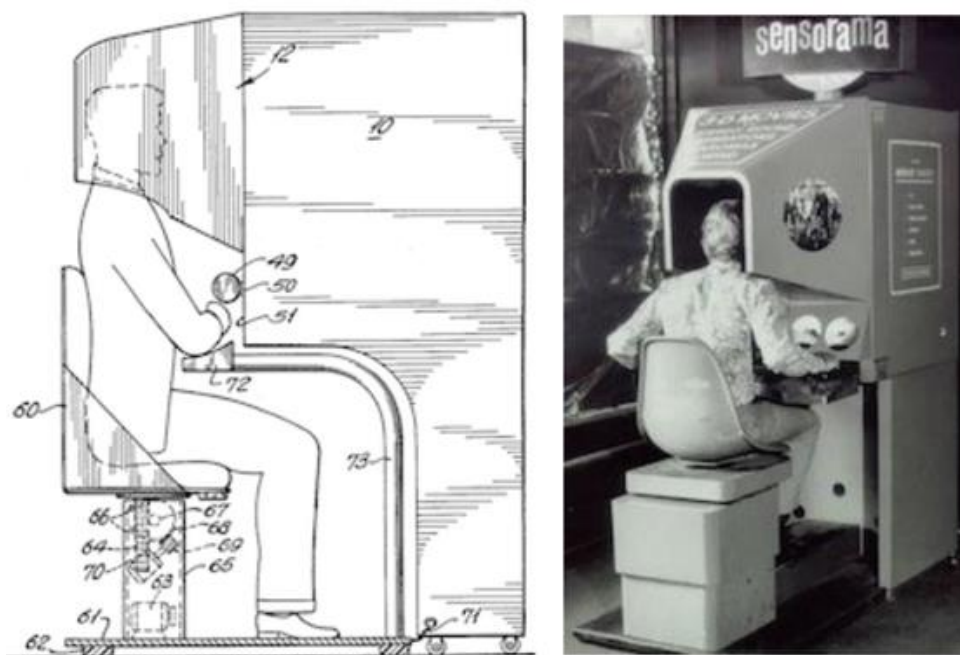
Nhìn chung, hệ thống hiển thị thực tế ảo tổng quát gồm ba phần chính: Phần mềm, thiết bị đầu vào và thiết bị đầu ra:

- **Phần mềm:** Là linh hồn của AR/VR và cũng như bất kỳ một hệ thống công nghệ nào. Phần mềm bao gồm ngôn ngữ lập trình, tạo hình 3D và các mô phỏng hành động. Đồng thời phần mềm là nơi tiếp nhận tương tác và phản hồi với những tác động của người dùng.
- **Thiết bị đầu vào:** Bao gồm những thiết bị cảm biến có khả năng ghi nhận những cử chỉ và hành động của người dùng như: Đầu, mắt, găng tay, chuột,... Riêng đối với AR thì không cần cảm biến, vì người dùng có thể tương tác với thiết bị ghi hình có camera.
- **Thiết bị đầu ra:** Bao gồm thiết bị có khả năng kích thích các giác quan để tạo cảm giác về sự hiện hữu trong thế giới ảo bao gồm các thành phần như: Hiển thị đồ họa, âm thanh nổi, bộ phản hồi cảm giác, bộ phản hồi xung lực,... tăng khả năng tương tác “thật” với người dùng như hành động thực tế. Riêng với hệ thống AR khả năng chỉ hiển thị phản hồi đồ họa với người dùng thông qua thiết bị ghi hình có camera.

1.2. Lịch sử hình thành và phát triển

Công nghệ thực tế ảo đã trải qua 50 năm nghiên cứu và phát triển để đi đến khả năng ứng dụng như ngày nay nhưng nó vẫn được xem là một công nghệ mới vì khả năng chiếm lĩnh thị trường chưa thực phổ biến.

Công nghệ thực tế ảo được phát minh vào năm 1957 bởi Morton Heilig. Thiết bị đa phương tiện của ông được gọi là Sensorama (Hình 1.1) cho phép người dùng xem video 3D và đắm mình vào cảm giác thực mà nó mang lại. Đồng thời ông cũng phát triển thiết bị HMD đầu tiên gọi là Telesphere Mark cho phép người dùng xem hình ảnh 3D theo kịch bản dựng sẵn [3][5].



Hình 1.1. Máy mô phỏng thực tế ảo Sensorama (nguồn Google)

Năm 1961, Comeau và Bryan đã phát triển thiết bị HMD gọi là Headsight, có hai màn hình hiển thị và theo dõi chuyển động của mắt. Năm 1968, giáo sư và là một nhà khoa học máy tính tại Harvard, Ivan Sutherland đã phát minh ra HMD có tên gọi là “The Sword of Damocles” có khả năng hiển thị các vật mẫu 3D đơn giản trước mắt người dùng trong không gian thực [3][5].

Năm 1987, John Lanier nhà khoa học, nhà nghiên cứu đã định ra thuật ngữ “thực tế ảo”, sáng lập ra phòng nghiên cứu ảo và cho thành lập công ty kinh doanh kính VR [3][5].

Năm 1995, Nintendo Virtual Boy cho ra mắt máy chơi game hỗ trợ không gian ba chiều. Năm 1997, các nhà nghiên cứu của Đại học Georgia Tech và Emory

đã hợp tác để tái hiện các khu vực chiến tranh tại Việt Nam để hỗ trợ các buổi tri liệu cho các cựu chiến binh [3][5].

Năm 2007, Google cho ra mắt chế độ xem phố trong Google Map, các hình ảnh toàn cảnh được chụp bằng một camera được cấp bằng sáng chế trên một chiếc xe chuyên dụng. Camera chụp các hình ảnh xung quanh xe khi nó đang di chuyển.

Năm 2014, Sony ra mắt dự án PlayStation VR và khuyến khích người chơi, không chỉ chơi trò chơi mà còn “trực tiếp” chơi trò chơi. Năm 2016, Google giới thiệu Cardboard một trình xem lập thể có thể tự chế tạo, giải pháp thay thế cho các thiết bị HMD đắt tiền.

Năm 2018, Một cuộc hội nghị tại Facebook, tiết lộ họ có thể cho ra mắt trong thời gian gần một chiếc HMD với tầm nhìn 140 độ (để đạt đến độ chân thực cần thiết, mắt người có tầm nhìn khoảng 120 độ mỗi bên).

Nhưng vậy có thể nói, quá trình phát triển của công nghệ thực tế ảo đang ngày càng cố gắng đưa giá trị ảo hướng dần đến hiện tại. Trải nghiệm của người dùng ngày càng cải thiện và chân thật hơn, ảnh hưởng rõ ràng hơn đến nhận thức của con người về thế giới.

Công nghệ thực tế ảo còn phát triển mạnh hơn nữa và có nhiều công bố gây bất ngờ đối với các nhà làm công nghệ trong tương lai tới.

1.3. Ứng dụng công nghệ thực tế ảo vào các lĩnh vực

Trong thời đại công nghệ 4.0, song song với sự nổi trội của “trí tuệ nhân tạo”, công nghệ thực tế ảo đang vươn mình phát triển trên nhiều lĩnh vực dường như không tưởng, có thể nói khả năng đa dạng ngành nghề đem đến những trải nghiệm tuyệt đối với người dùng:

- **Y tế, huấn luyện:** Áp dụng các kỹ thuật giả lập trong môi trường ảo để nghiên cứu thực nghiệm bằng các chuyển động mô phỏng cơ quan nội tạng nhằm để nghiên cứu và thực hành nghiệp vụ trước khi phẫu thuật trên cơ thể thật [6][7].
- **Giáo dục:** Có thể tái hiện các lớp học lịch sử, các trạm nghiên cứu, các môi trường ngoài tự nhiên, vũ trụ mà học sinh, sinh viên không có điều kiện tiếp xúc ngoài thực tế. Mô phỏng và trải nghiệm thực tế, tăng khả năng tiếp thu của sinh viên hơn sách giáo khoa [7].
- **Ngành thời trang may mặc, thiết kế, buôn bán bất động sản:** Khả năng tái tạo trước khi quyết định có mua bán với sản phẩm đã trình bày hay

không như: Thử quần áo, sắp đặt nội ngoại thất, mô phỏng mảnh đất quy hoạch,... [6].

- **Truyền thông quảng cáo:** Một trong những thế mạnh của công nghệ thực tế ảo, nhằm thu hút trải nghiệm của người dùng. Các công ty, doanh nghiệp lớn đưa ra những chiến lược kinh doanh táo bạo, áp dụng công nghệ AR nhằm đưa các sản phẩm “ảo” hiện hữu ra thế giới thực. Tạo ra các câu chuyện của thương hiệu trong môi trường ảo [6].
- **Trong quân sự:** Tạo ra các mô phỏng các khu vực chiến lược, tập trận giả, thiết lập các cuộc phỏng binh, thực hành điều khiển các thiết bị quân sự chuyên dụng,... [7].

1.4. Vấn đề an toàn trong sử dụng công nghệ thực tế ảo

Bên cạnh những tác động tích cực mà công nghệ thực tế ảo đem lại, với khả năng có thể đem những giá trị ảo ra bên ngoài thế giới thực. Người dùng có thể quá lệ thuộc vào những giá trị tinh thần mà nó đem lại dẫn đến những tình trạng rối loạn tâm lý, ảnh hưởng đến mắt nếu như lạm dụng [8].

Việc người sử dụng thiết bị thực tế ảo để xem mô phỏng ảo, sẽ không quan tâm đến những sự kiện xảy ra xung quanh, có thể dẫn đến những tai nạn khó lường trước được. Bên cạnh những ảnh hưởng về mặt tâm lý là ảnh hưởng về mức độ kỹ thuật và dữ liệu người dùng. Thêm một kiểu dữ liệu cá nhân mới là thêm một rủi ro về quyền riêng tư [8].

Các ứng dụng AR/VR đều yêu cầu người dùng cung cấp quyền sử dụng camera, một vài yêu cầu quyền truy cập địa điểm hiện tại, các ứng dụng IoT (Internet kết nối vạn vật) thu thập các cử chỉ và tương tác người dùng để tái sử dụng trong tương lai. Các thiết bị như kính chụp đầu – HMD, ghi lại các tương tác người dùng như: Cử chỉ mắt, thao tác đầu, cổ,... [9].

Đây là một thách thức đối với các công ty cung cấp AR/VR nhằm bảo vệ dữ liệu người dùng mà họ thu thập. Đối với các dạng dữ liệu khác, các công ty này phải minh bạch về cách họ lưu trữ, thu thập và sử dụng dữ liệu người dùng hay cách họ chia sẻ với bên thứ ba về dữ liệu họ thu thập [8][9].

Về phía người dùng, phải chắc chắn rằng dữ liệu mình được sử dụng bởi các nhà cung cấp dịch vụ đáng tin cậy. Giả sử trong trường hợp bị lợi dụng, kẻ tấn công có thể thay thế hiển thị đồ họa và các thông số truyền đạt để đưa thông tin sai lệch đến cho người dùng [8][9].

Ngoài ra, sự phát triển của công nghệ giấu tin trong ảnh, giấu tin trong các thực thể 3D dẫn đến khả năng thông tin có thể bị giấu theo cách này. Tùy theo mục đích sử dụng, nếu kẻ giấu tin có mục đích vụ lợi cá nhân, có khả năng ảnh hưởng lớn đến uy tín của một tổ chức hoặc người sử dụng.

1.5. Cơ bản về kỹ thuật giấu tin

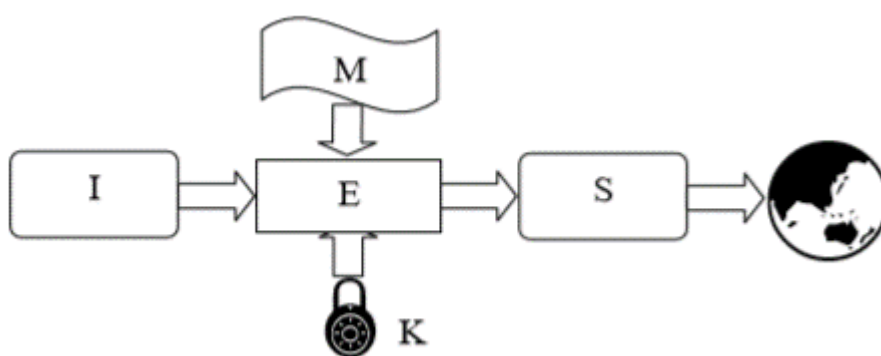
Kỹ thuật giấu tin là nghệ thuật che giấu một lượng thông tin, được quy ra dưới dạng chuỗi bit vào một đối tượng có thể chứa thông tin. Có nhiều phương tiện được sử dụng cho việc che giấu, mang thông tin như: Ảnh, âm thanh, video, một đoạn văn bản,... Vật chứa (cover) khi mang thông tin được giấu gọi là vật mang tin (stego).

Vật chứa thông tin có thể không thay đổi kích thước hay đặc tính khi trở thành vật mang tin. Đặc điểm nổi bật là cả người gửi lẫn người nhận đều khó nhận biết được thông tin đã bị giấu vào trong đối tượng.

Hiện nay, giấu tin trong ảnh chiếm tỉ lệ lớn nhất trong hệ thống đa phương tiện. Tuy nhiên, trường hợp giấu tin áp dụng công nghệ VR/AR không phải trường hợp không xảy ra.

Quá trình giấu tin được chia thành hai khối có cấu trúc khác nhau: Quá trình nhúng tin và quá trình giải mã:

- **Quá trình nhúng tin:** Đầu vào là đối tượng chứa thông tin (M) và dữ liệu cần che giấu (I). Đầu ra là đối tượng mang thông tin đã được nhúng (S) và khóa bí mật (K). Đối tượng mang và khóa bí mật được gửi cho người nhận qua một kênh truyền an toàn [1][14].

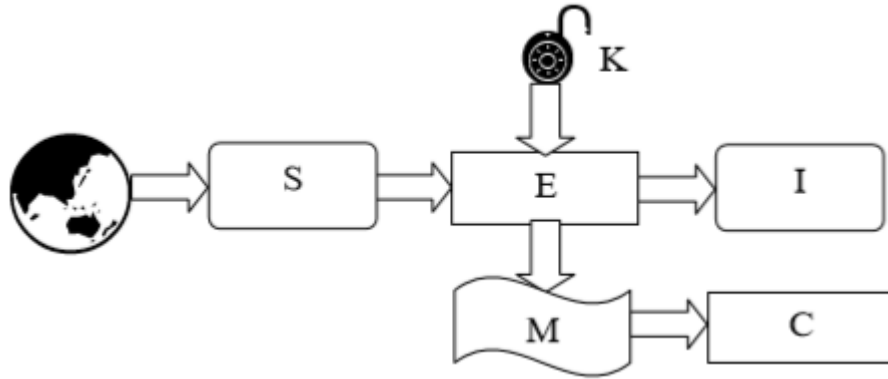


Hình 1.2. Lược đồ chung cho quá trình giấu thông tin

Trong lược đồ trên các kí hiệu mang ý nghĩa biểu thị như sau:

- M: Thông tin cần giấu.
- I: Dữ liệu phủ, môi trường giấu tin.
- E: Bộ mã hóa/ giải mã, chương trình – thuật toán nhúng tin.

- K: Khóa bí mật sử dụng cho quá trình mã hóa và giải mã.
- S: Dữ liệu mang tin mật.
- **Quá trình trích xuất:** Sử dụng đối tượng mang và quy tắc khóa bí mật mà người gửi đã gửi để người nhận có thể tìm ra dữ liệu đã che giấu [1][14].



Hình 1.3. Lược đồ chung cho quá trình giải mã thông tin

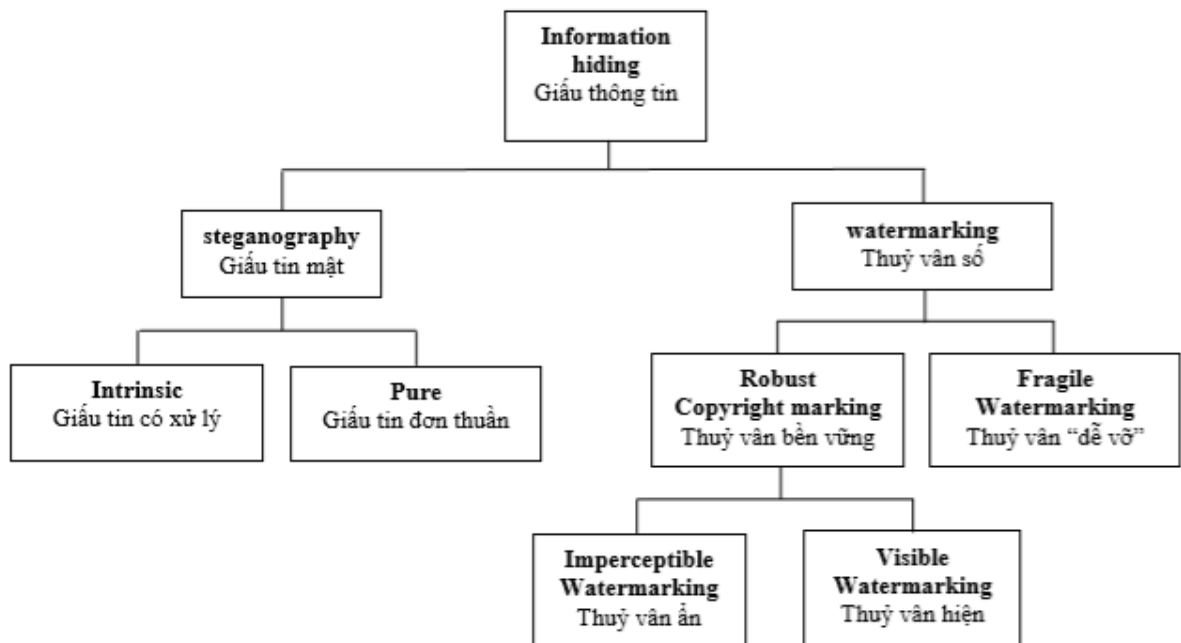
Trong lược đồ trên các kí hiệu mang ý nghĩa biểu thị như sau:

- M: Thông tin cần giấu.
- I: Dữ liệu phủ, môi trường giấu tin.
- E: Bộ mã hóa/ giải mã, chương trình – thuật toán nhúng tin.
- K: Khóa bí mật sử dụng cho quá trình mã hóa và giải mã.
- S: Dữ liệu mang tin mật.
- C: Kiểm định, nhằm kiểm tra thông tin sau khi giải mã, có ý nghĩa không. Đối với thủy vân số, văn bản được kiểm tra có giống với thủy vân được ký trên đó không.

1.6. Phân loại kỹ thuật giấu tin

Phạm trù về việc ẩn giấu dữ liệu được chia thành rất nhiều nhánh khác nhau: Theo Hình 1.4, có 2 đặc tính thường được áp dụng để che giấu thông tin phổ biến là giấu tin mật (steganography) và thủy vân số (watermarking) [1][15]:

- **Kỹ thuật giấu tin mật:** Với mục đích đảm bảo an toàn cho thông tin được giấu. Tập trung vào việc giấu thông tin được nhiều nhất có thể nhưng khó phát hiện ra thông tin trong đối tượng mang tin [1][15].
- **Thủy vân số:** Với mục đích bảo mật cho chính đối tượng mang tin. Đảm bảo tính bền vững, khẳng định bản quyền sở hữu, chống xuyên tạc, lệch lạc thông tin gốc. Không quan trọng số lượng thông tin ẩn giấu. Thủy vân có thể được nhìn thấy hoặc không đối với vật mang tin [1][16][18].



Hình 1.4. Lược đồ phân loại các phương pháp giấu tin

Kỹ thuật giấu tin trong ảnh làm tiền đề để hình thành nên các kỹ thuật giấu tin trong các phương tiện khác nhau. Trong giới hạn bài nghiên cứu, sẽ không trình bày phương pháp thủy văn số [18].

Một số phương pháp giấu tin trong ảnh được áp dụng trong nghiên cứu và thực nghiệm:

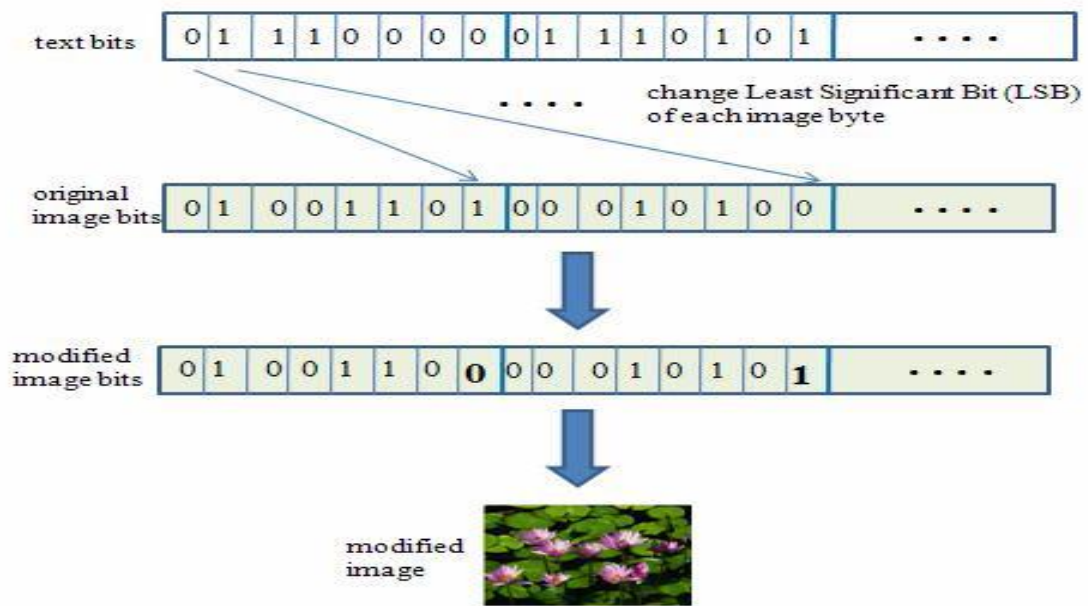
1.6.1. Kỹ thuật giấu tin LSB

LSB (Least Signification Bit): Là bit có ảnh hưởng ít nhất đến việc quyết định tới màu sắc của mỗi điểm ảnh [1][17].

Kỹ thuật giấu tin LSB là phương pháp giấu tin đơn giản nhất, thông điệp dưới dạng nhị phân sẽ được giấu (nhúng) vào các bit có trọng số thấp nhất.

Ví dụ: Đối với ảnh màu 16 bit thì 15 bit là biểu diễn ba màu RGB của điểm ảnh, còn một bit cuối không dùng đến sẽ được tách ra thành một bit ở mỗi điểm ảnh để giấu tin.

Trong phép tách này, việc thay đổi giá trị của bit cuối cùng dẫn đến việc thay đổi giá trị của điểm ảnh lên hoặc xuống đúng một đơn vị. Với sự thay đổi nhỏ đến thì cấp độ màu của điểm ảnh sẽ không bị thay đổi nhiều (Hình 1.5) [1][17].



Hình 1.5. Thay đổi giá trị thứ yếu của điểm ảnh để che giấu chuỗi bit

1.6.2. Kỹ thuật giấu tin Wu-Lee

Thuật toán này được đưa ra bởi M.T.Wu và L.H.Lee [1][18].

Với đối tượng mang tin là một ảnh đen trắng, được phân tích thành các ma trận điểm ảnh với các giá trị 0 và 1.

Ý tưởng của thuật toán là làm thế nào để nhúng được vào nhiều khối, mỗi khối chỉ một bit thông tin. Nếu khối ma trận thỏa mãn một điều kiện định trước thì phần tử của khối ma trận đó được thay đổi để chứa đựng thông tin.

Do thuật toán chỉ làm thay đổi nhiều nhất là một bit thông tin trong mỗi khối nên khi kích thước của khối ma trận đủ lớn thì việc thay đổi một trong các khối ma trận là điều không đáng kể.

Ngoài ra, còn một số kỹ thuật giấu tin trên ảnh khác như: Kỹ thuật giấu tin Chen – Pan – Tseng (phát triển từ kỹ thuật Wu – Lee, khả năng cải tiến giấu tin trong ảnh màu), kỹ thuật giấu tin dựa trên bảng màu, kỹ thuật trải phổ, kỹ thuật dùng hệ số DCT,... [18]

Như vậy, giấu tin áp dụng hai phương pháp chính là: Giấu tin mật và thủy vân số để đảm bảo an toàn cho nguồn tài nguyên được che giấu. Trong thực tế, có nhiều cách thức khác nhau để che giấu thông tin. Che giấu thông tin trên ảnh với kỹ thuật LSB là kỹ thuật cơ bản nhất và là cơ sở để đặt nền móng nâng cao cho các phát kiến sau này.

1.7. Phương pháp giấu tin trong các đối tượng đồ họa

Hiện nay, không chỉ hình ảnh hay thông điệp phẳng được truyền thông rộng rãi, bên cạnh đó là các sản phẩm đồ họa khác nhau được biết đến như: Các đối tượng vật thể ba chiều, video, các đối tượng đồ họa thực tế ảo,...

Về cơ bản chung quy phương pháp xử lý các đối tượng trên là giấu tin đối với vật chứa tin là hình ảnh. Các đối tượng đồ họa, chỉ khác nhau về phương thức thể hiện đối với người dùng nhưng nhìn chung bản chất là các khung hình chuyển động hoặc lập thể.

1.7.1. Phương pháp giấu tin trong thực thể ba chiều

Trong không gian, các đối tượng đa giác bao gồm các thành phần khác nhau như: Các đỉnh, các cạnh, các mặt, các đa giác,... Trong thực tế, các đối tượng ba chiều không hề đơn giản đến như vậy mà biến đổi theo nhiều kiểu cấu trúc để đạt hiệu quả cao nhất về mặt hình ảnh. Tuy nhiên, để biểu diễn các đối tượng hình ảnh phức tạp, người ta dùng mô hình đối tượng lưới [12][13].

Trong mô hình đối tượng lưới, các mặt tam giác luôn là hình được sử dụng nhiều nhất, các hình khối ba chiều kiểu này được gọi là low poly, có khả năng tối ưu hóa hiển thị. Số lượng các mặt đa giác càng nhiều thì hình ảnh càng rõ nét (high poly). Điều này ảnh hưởng ít nhiều đến dung lượng và tốc độ xử lý ảnh [13]. Từ nghiên cứu trên, đưa ra được định nghĩa cho các biểu diễn lưới tam giác:

Định nghĩa: Cho tập đỉnh $V = [V_1, V_2, \dots, V_n]$ với mỗi đỉnh là một bộ ba các giá trị tọa độ x, y, z trong không gian, n là tổng số đỉnh. Một biểu diễn lưới tam giác trong không gian ba chiều là một tập cấu trúc lưu trữ thông tin về kết nối các đỉnh [12].

$$I = \{ I_1; I_2; \dots; I_k \}$$

Với $1 \leq k \leq n$. I_i (với $1 \leq i \leq k$) là bộ 3 các chỉ số (u, v, t) với $1 \leq u < v < t \leq n$.

Ví dụ: Cho tập $V = [V_1, V_2, V_3, V_4]$

- Hình chóp C có thể được biểu diễn dưới dạng lưới (Hình 1.6.a):

$$I_C = \{(1,2,3);(1,2,4);(1,3,4);(2,3,4)\}$$

- Hình 1.6.b biểu diễn lưới tam giác:

$$I_M = \{(1,3,4);(2,3,4)\}$$



Hình 1.6. Mô hình biểu diễn lưới tam giác

Ứng với khả năng xây dựng mô hình lưới tam giác, để giấu tin trong miền không gian này, một kỹ thuật giấu tin được đề xuất là phương pháp MEP. Hay được gọi với tên là kỹ thuật giấu tin trên tam giác. Bao gồm các quá trình:

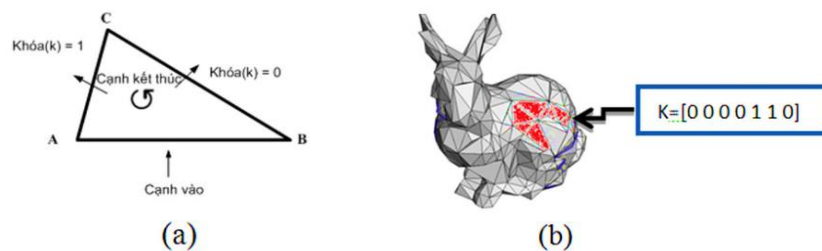
▪ **Xây dựng tam giác:**

Trong một tam giác ở mô hình lưới, định nghĩa các đặc tính như:

- Cạnh vào là cạnh dùng để đi vào tam giác
- Hai cạnh kết thúc để đi tới tam giác kế tiếp

Ví dụ: Giả sử, (Hình 1.7.a) tam giác ABC, với AB là cạnh vào thì khi đó, AC và BC lần lượt là cạnh ra của tam giác đó. Ứng với khóa K là 1 hoặc 0 thì cạnh vào tương ứng của tam giác tiếp theo là cạnh đã được quy ước.

Nếu giá trị bit khóa (tại giá trị đang xét) bằng 1, tam giá kế tiếp có cạnh vào là BC. Ngược lại bằng 0 thì có cạnh vào là AC.



Hình 1.7. Phương pháp xác định tam giác kế tiếp dựa trên khóa nhị phân

Như vậy, cạnh kết thúc của tam giác này là cạnh vào của tam giác kế tiếp. Hình 1.7.b cho thấy hiển thị danh sách các tam giác được duyệt với chuỗi bit khóa phát sinh. Số các tam giác được duyệt bằng số độ dài chuỗi bit khóa. Trong trường hợp giấu M bit, ứng với mỗi đỉnh của tam giác là một bit, số bit khóa tương ứng là: $M/3$ [12].

▪ **Giấu tin trong tam giác:**

Thay đổi thông tin nhờ vào tham chiếu của Một đỉnh kết thúc của tam giác lên cạnh vào của tam giác đó. Cạnh vào được chia thành các tập con được biểu

diễn luân phiên thông điệp. Để nhúng bit thứ i ($i = 0$ hoặc 1) vào đỉnh kết thúc. Cần kiểm tra xem tham chiếu của nó lên cạnh vào có tương ứng không, nếu không tương ứng với giá trị cần nhúng, dịch chuyển đỉnh kết thúc qua một trục đối xứng của cạnh vào [12].

Giá trị khoảng phân đoạn quyết định mức độ dịch chuyển của tam giác, sao cho dịch chuyển phải đủ nhỏ để không làm thay đổi hình dạng ban đầu. Các thay đổi này có thể làm thay đổi đến cục diện của đối tượng. Nếu số lượng của lưới tam giác là vô cùng lớn thì thay đổi của vật thể mang tin là không đáng kể.

1.7.2. Phương pháp giấu tin trong video

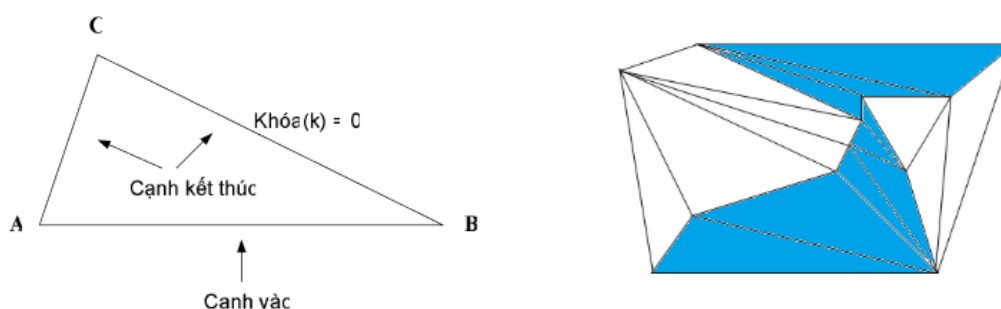
Giấu tin trong video cũng sử dụng phương pháp giấu tin trong bit có trọng số thấp (LSB) và mang hiệu quả tương tự như giấu tin trong ảnh, chỉ khác là giấu tin qua nhiều miền khung hình [18].

1.8. Khả năng ẩn giấu và tương tác thông tin sử dụng thực tế ảo

Giấu tin sử dụng công nghệ thực tế ảo quy về bài toán ẩn tin trong các đối tượng ba chiều. Thử thách đặt ra là khả năng tương tác của người dùng không được làm thay đổi thông tin ẩn giấu. Vì trong các đối tượng trong môi trường ảo, có thể bị lệ thuộc vào tương tác của người dùng bao gồm: Góc nhìn, luật xa gần, đối với AR còn mang khả năng tương tác với môi trường ngoài,... [10]

Phương pháp thường được sử dụng để giấu tin trong các đối tượng thông tin ba chiều có thể kể đến mô hình lưới tam giác ba chiều. Đối với giấu tin trong mô hình ba chiều, đối tượng tiếp xúc có thêm một nhân tố quyết định đến độ nổi trong không gian là trục z . Các thông tin được mã hóa ở dạng bit và sau đó được đưa vào đối tượng ở dạng điểm ảnh [11][12].

Về kết cấu của đối tượng 3D, ở dạng tối ưu dung lượng hay được biết đến với thuật ngữ low poly hay polygon là dạng đối tượng thể hiện bởi các mặt cấu trúc đa giác, khả năng tiết kiệm tài nguyên tối đa. Đa số các bề mặt cấu trúc được cấu thành từ các bề mặt tam giác. Từ đây thông tin có thể được ẩn giấu trên các bề mặt này theo mô hình lưới tam giác ba chiều [13].



Hình 1.8. Phương pháp xây dựng tam giác giấu tin và mô hình lưới ba chiều

1.9. Kết luận chương 1

Công nghệ thực tế ảo đã có hơn 50 năm phát triển, ứng dụng hầu hết trong các lĩnh vực hiện đại. Không dừng ở thực tế ảo và thực tế ảo tăng cường, các nhà phát triển đang cố gắng đưa người dùng đến với trải nghiệm chân thật nhất đối với các thông tin từ thế giới ảo. Tuy nhiên, cũng không tránh được nhiều vấn đề liên quan đến an toàn thông tin như: Thu thập thông tin người dùng, an toàn lưu trữ, truy cập dữ liệu và đặc biệt là khả năng tích hợp giấu tin vào ứng dụng thực tế ảo – một vấn đề ít được các nhà phát triển quan tâm hơn lợi ích nhận được từ chúng.

Kỹ thuật giấu tin có nhiều phương pháp nghiên cứu, đem lại hiệu quả khác nhau. Việc tìm hiểu rõ quá trình nhúng và giải mã giúp xây dựng hệ thống ẩn tin tốt hơn. Kỹ thuật giấu tin vào các bit thứ yếu (LSB) và kỹ thuật Wu - Lee có đối tượng tiếp cận là bit – giúp cho thông tin thay đổi khó nhận diện trong môi trường triển khai. Phương pháp giấu tin trong lưới tam giác ba chiều và giấu tin trong video là những phương pháp hứa hẹn mang lại hiệu quả cao để phát triển giấu tin bằng công nghệ thực tế ảo.

Hiểu rõ về hệ thống thông tin trong thực tế ảo và các cơ chế giấu tin hiện đại, giúp quá trình triển khai tránh được những thiếu sót trong quá trình vận hành tại môi trường ngoài thử nghiệm, một cách an toàn hơn.

CHƯƠNG 2. PHÂN TÍCH, THIẾT KẾ MÔI TRƯỜNG THỰC TẾ ẢO CHO ỨNG DỤNG

2.1. Giới thiệu yêu cầu ứng dụng

Trên khả năng các ứng dụng thực tế ảo ngày càng chiếm đại đa số các sản phẩm công nghệ trên thị trường, bởi khả năng chiếm lĩnh không gian của chúng. Với sự hỗ trợ mạnh mẽ của các thư viện xử lý đồ họa, công nghệ nhận diện và các tác động hành vi của người dùng mà thực tế ảo dần trở thành xu hướng hóa.

2.1.1. Yêu cầu của ứng dụng thực tế ảo tăng cường (AR)

Thực tế ảo tăng cường, cho phép khả năng đưa thế giới ảo ra ngoài môi trường thật. Về khả năng ẩn giấu thông tin có thể thực hiện được do khả năng tích hợp nhiều mô hình hiển thị dữ liệu ra môi trường ngoài cho người dùng có thể thấy được bằng thiết bị ghi hình (ví dụ: Điện thoại thông minh, cảm biến quét hình ảnh...) [2][6].

▪ **Yêu cầu cơ bản:**

- Hiển thị vật thể ba chiều với nền đơn cho các thiết bị không được hỗ trợ công nghệ thực tế ảo tăng cường (AR). Có khả năng tương tác được. Vật mang tin không bị thay đổi về đặc tính nguyên thủy.
- Hỗ trợ bật tắt chế độ AR cho các máy có hỗ trợ công nghệ thực tế ảo tăng cường (AR). Yêu cầu quyền truy cập khi khởi động ứng dụng.
- Cho phép hiển thị và tương tác với vật thể trong không gian qua thiết bị ghi hình. Hỗ trợ đa góc nhìn và ghi nhớ vật thể trong không gian.
- Chỉ dẫn người dùng về khả năng phát hiện bề mặt đặt vật thể trong không gian và có cơ chế thông báo kèm theo.
- Đảm bảo thu thập dữ liệu người dùng được lưu trữ và vận chuyển thông qua kênh an toàn, mã hóa và bảo mật dữ liệu khi người dùng chia sẻ: Vị trí, vị trí ẩn giấu thông tin, vị trí ẩn giấu vật thể trong thế giới thật, số lần tương tác với vật thể...

▪ **Khả năng:** Không chỉ đơn giản là hiển thị vật thể ba chiều ra không gian thực mà thực tế ảo tăng cường có thể thực hiện những đề mục sau:

- Hiển thị vật thể ba chiều có chuyển động trong thế giới thực.
- Thực hiện ghi nhớ vật thể trong không gian thực theo thời gian bằng công nghệ Google Anchor Clouds và GPS.

- Cho phép hiển thị đa phương tiện trên một hình ảnh mẫu nhận dạng trước như: Đối tượng ba chiều, hình ảnh khác, video,... (Augmented Images – Thực tế ảo tăng cường qua hình ảnh).
- Khả năng bán kết hợp giữa AR và VR thông qua (Augmented Portal – Thực tế ảo tăng cường liên kết không gian ảo) cho phép người dùng bước qua thế giới ảo thông qua AR và thiết bị ghi hình.
- Một khả năng thường thấy trong các ứng dụng nhận diện khuôn mặt hiện nay là Augmented Faces. Cho phép người dùng thêm các hiệu ứng hay vật thể chuyển động ứng với chuyển động mặt.

Như đã trình bày về khả năng mở rộng của các ứng dụng thực tế ảo tăng cường là rất cao do phạm vi tích hợp đa phương tiện là vô cùng lớn. Kèm theo đó là một vài thách thức đối với khả năng ẩn giấu thông tin sử dụng AR.

▪ **Thách thức:**

- Đối tượng ba chiều trong không gian thực không thực sự là một đối tượng tĩnh, nên việc ẩn giấu thông tin phải đòi hỏi sự phân tích kỹ lưỡng đối với tập tin chứa thông tin về đối tượng đó để không làm xáo trộn hay làm ảnh hưởng đến chất lượng cuối.
- Việc thông tin ẩn có thể bị mất do không lưu lại vị trí trong thế giới thật hoặc bị biến dạng thông qua kênh truyền đến tay người nhận.
- Tích hợp tính năng Augmented Portal phải thông qua công nghệ Virtual Reality vì khả năng Augmented Reality khó có thể phân tích hay xử lý một bối cảnh ảo trên diện rộng được.

2.1.2. Yêu cầu của ứng dụng thực tế ảo (VR)

Thực tế ảo tăng cường, khả năng cho phép người dùng trải nghiệm tương quan trong thế giới ảo. Về khả năng ẩn giấu thông tin trong các đối tượng ba chiều, các bối cảnh đặt vật thể có thể hiển thị được thông qua kính thực tế ảo. Đơn cử video 360 có thể chạy trên các thiết bị di động, sử dụng kính thực tế ảo như công cụ hỗ trợ để đem lại cảm giác chân thật cho người trải nghiệm, đây cũng được xem như một dạng của công nghệ VR đời đầu [4][6].

▪ **Yêu cầu cơ bản:**

- Cho phép theo dõi chuyển động (tracking) thiết bị người dùng (như: Chuyển động đầu với phần mềm VR được nạp sẵn, chuyển động cử chỉ tay, hay chuyển động thiết bị di động với các góc quay khác nhau...).

- Yêu cầu quyền truy cập trước khi khởi động ứng dụng đối với các ứng dụng VR hỗ trợ trên nền tảng di động.
- Cho phép hiển thị và tương tác với vật thể trong không gian qua thiết bị ghi hình. Hỗ trợ đa góc nhìn và ghi nhớ vật thể trong không gian.
- Chỉ dẫn người dùng về khả năng tương tác với vật thể trong không gian và có cơ chế thông báo kèm theo.
- Đảm bảo thu thập dữ liệu người dùng được lưu trữ và vận chuyển thông qua kênh an toàn, mã hóa và bảo mật dữ liệu khi người dùng chia sẻ: Vị trí, vị trí ẩn giấu thông tin, vị trí ẩn giấu vật thể trong thế giới ảo, số lần tương tác với vật thể...
- **Khả năng:** Không chỉ đơn giản là hiển thị vật thể ba chiều ra không gian ảo mà công nghệ VR có thể thực hiện những đề mục sau:
 - Hiển thị vật thể ba chiều có chuyển động trong thế giới ảo. Dẫn dắt người dùng đến với trải nghiệm tương tác ảo như tương tác với thế giới thật. Một ứng dụng thành công, mang lại cảm giác gần gũi của người dùng như đang nhập vai vào thế giới/trò chơi ảo mang lại.
 - Dữ liệu của người dùng có thể được ghi lại và lưu trữ trên các hệ thống máy chủ riêng. Ghi nhận các thao tác, các hoạt động của người dùng trong thế giới ảo.
 - Hệ thống thế giới ảo online đang trên đà thử nghiệm và phát triển, cho phép nhiều người chơi từ nhiều nơi khác nhau tương tác với cùng một hệ thống phần mềm, cho phép nhiều trải nghiệm thú vị.
 - Công nghệ Mixed – Reality là sự kết hợp giữa AR và VR cho phép người dùng có đặt vật thể ảo bằng công nghệ AR trong môi trường ảo VR.

Bằng những ưu điểm cao về mặt chức năng và trải nghiệm, VR là một công nghệ thực tế ảo đáng trông đợi, nhưng song song đó VR gặp một số những cản trở kinh tế và thách thức ẩn giấu thông tin.

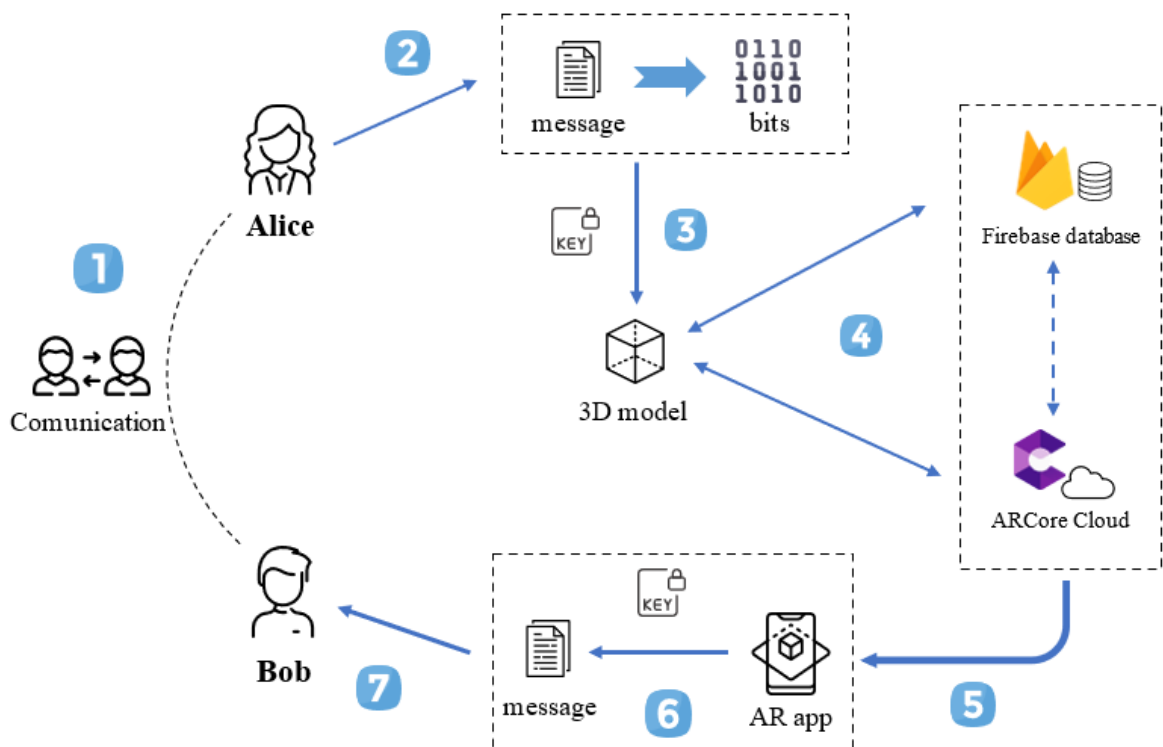
- **Thách thức:**
 - Đối tượng ba chiều trong không gian ảo, không thực sự là một đối tượng tĩnh. Vật thể trong thế giới ảo, có thể có sự kiện kèm theo hoặc đi kèm một bối cảnh riêng, nên việc phân tích hành vi tác động của người dùng để không bị ảnh hưởng chất lượng thông tin được che giấu.
 - Việc thông tin ẩn giấu có thể bị mất do không lưu lại vị trí trong thế giới ảo hoặc bị biến dạng thông qua kênh truyền đến tay người nhận.

- Tích hợp tính năng Augmented Portal phải thông qua công nghệ AR vì khả năng AR khó có thể phân tích hay xử lý một bối cảnh ảo trên diện rộng được.
- Giá thành các thiết bị hỗ trợ có chi phí thị trường khá cao, chất lượng tùy thuộc vào giá thành. Nên khả năng thu được thông tin và trải nghiệm tốt là không tương xứng.

2.2. Phân tích thiết kế mô hình ứng dụng

Từ những yêu cầu mang tính thử thách được đặt ra cho thấy được tầm ảnh hưởng sâu rộng của công nghệ thực tế ảo, cũng như khả năng giấu tin vào các vật thể đa chiều. Trên nền tảng lý thuyết về công nghệ thực tế ảo, ta thực hiện phân tích mô hình ứng dụng thử nghiệm trên nền tảng thiết bị di động (Android).

Trong thực tế không có quá nhiều mô hình triển khai cho AR/VR, một hệ thống đầy đủ các thành phần giám sát thường mang tính phức tạp và bí mật hơn [2][6]. Mô hình dưới đây trình bày hệ thống chia sẻ thông tin mật theo thời gian thực áp dụng công nghệ AR:



Hình 2.1. Mô hình che giấu thông tin áp dụng công nghệ AR và mật mã

Theo sơ đồ Hình 2.1, tại điểm (1) Alice và Bob có nhu cầu trao đổi thông tin mật với nhau, họ quyết định sử dụng một phần mềm AR để làm cổng giao tiếp thông tin giữa hai bên. Thông thường việc chia sẻ dữ liệu bản rõ trên đường truyền có thể

dễ dàng bắt được và giải mã; nhưng với hệ thống nhúng tin sử dụng vật mang là đối tượng ba chiều, điều này làm bài toán giải mã thông điệp trở nên khó khăn hơn đối với những truy cập bất hợp pháp.

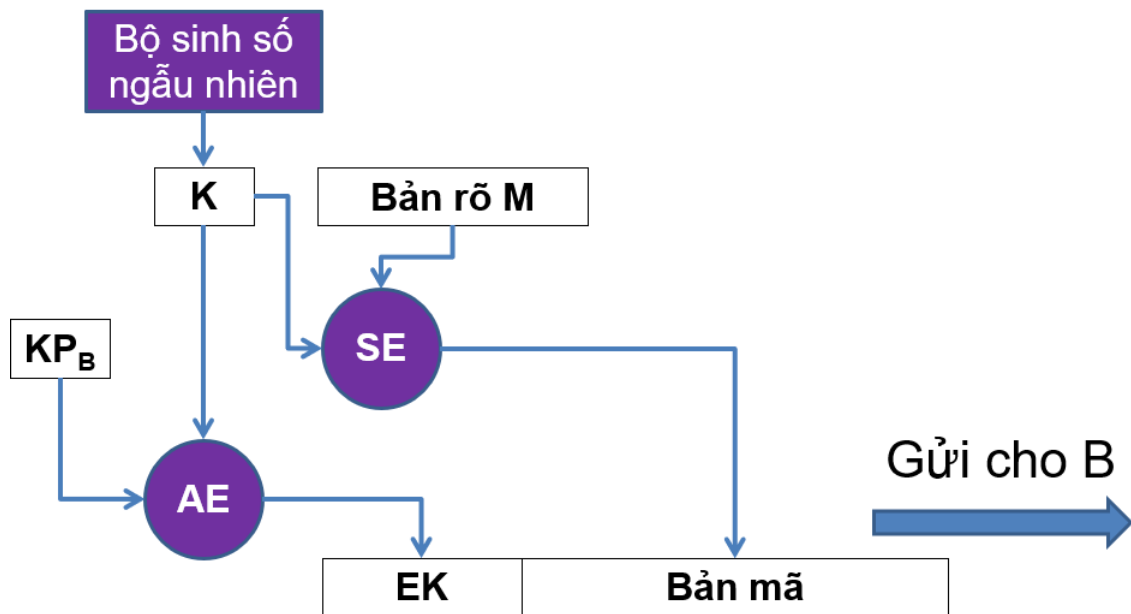
Để hiểu rõ hơn về các luồng xử lý dữ liệu, ta đi vào giải thích và phân tích các chức năng, vai trò của các thành phần trong hệ thống, cũng như các điểm dừng quan trọng (các điểm được đánh dấu trong mô hình).

▪ *Người dùng (thiết bị đầu – cuối end-to-end):*

- Alice và Bob đóng vai trò tương tác với hệ thống trong mô hình.
- Khi Alice có nhu cầu liên lạc với Bob và lựa chọn kênh truyền thông tin thông qua ứng dụng giấu tin AR, tại thời điểm có hoặc không có đồng người xung quanh; hoặc tại một thời điểm vắng mặt Bob như một lời nhắc nhở, một lời hẹn khi Bob có mặt tại vị trí đó.
- Alice và Bob tương tác với nhau thông qua ứng dụng, biết được tài khoản sử dụng của nhau và đã xác thực thông tin trước đó.

▪ *Chuyển đổi thông tin:*

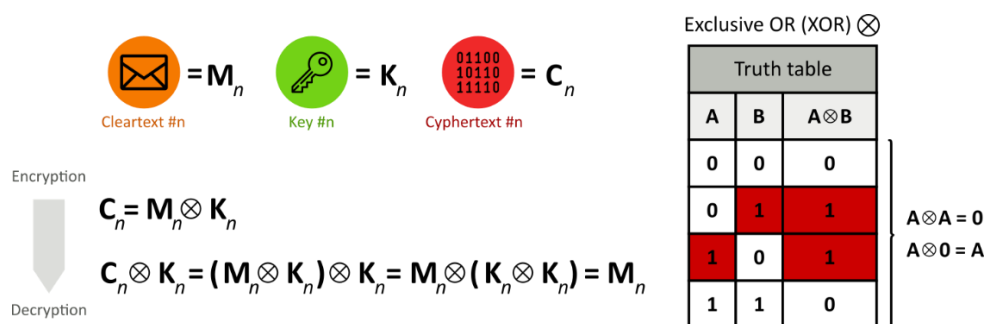
- Tại điểm dừng thứ (2) Alice nhập thông điệp của mình gửi đến Bob từ màn hình giao diện ứng dụng. Thông tin được chuyển đổi trực tiếp tại thiết bị người dùng hoặc được xử lý tại máy chủ (tùy chỉnh cấu hình) thông qua hàm chuyển đổi thông điệp thành dạng bit nhị phân.
- Để đảm bảo tính bí mật, mỗi phiên sử dụng của người dùng. Ứng dụng cung cấp cho phiên chia sẻ dữ liệu một khóa phiên. Khóa phiên sử dụng thuật toán AES cùng với dữ liệu đầu vào mã hóa thành bản mã [19].
- Thông điệp của Alice gửi cho Bob được mã hóa bằng khóa phiên. Song song đó, khóa phiên đóng vai trò khóa bí mật của quá trình trao đổi được mã hóa với khóa công khai của Bob [19].
- Khóa phiên và thông điệp sau khi được mã hóa được đưa về dạng bit nhị phân. Khóa phiên có độ dài xác định được nối với chuỗi thông điệp nhằm xác định chính xác giới hạn của khóa bí mật và thông điệp. Cả khóa phiên và thông điệp đã mã hóa đều được gửi cho Bob. (Mô phỏng mã hóa: Hình 2.2) [19].
- Mỗi khóa được sinh ra nhờ vào bộ sinh khóa giả ngẫu nhiên. Cặp khóa bất đối xứng được sinh ra mỗi khi người dùng đăng nhập hoặc tạo tài khoản mới. Các khóa này được lưu trữ trên cơ sở dữ liệu đám mây của ứng dụng.



Hình 2.2. Mô hình mã hóa đối xứng và bất đối xứng kết hợp

▪ **Vật mang thông tin:**

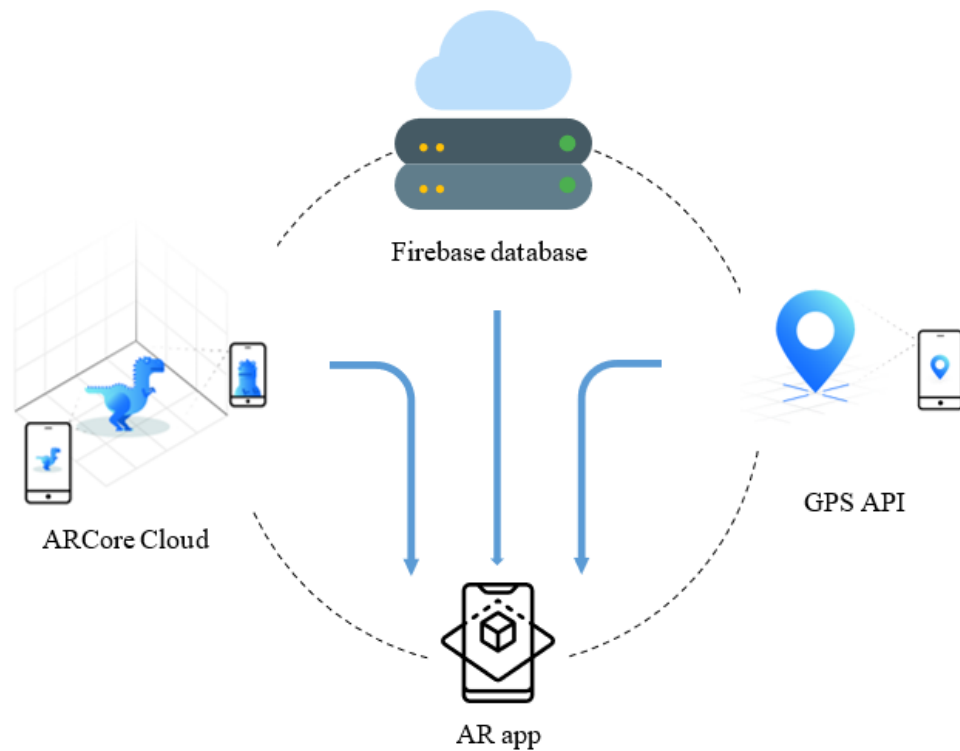
- Thông tin sau khi được mã hóa và chuyển đổi thành bit nhị phân. Tại điểm (3), Alice lựa chọn đối tượng ba chiều – đóng vai trò là vật mang tin để gửi cho Bob.
- Đối tượng ba chiều có định dạng đặc biệt, tồn tại tệp tin bit nhị phân mang thông tin đối tượng; được lợi dụng là môi trường chứa thông tin.
- Tại điểm (2), khóa phiên và thông điệp sau khi được mã hóa được chuyển thành chuỗi bit nhị phân. Sang điểm (3), khi Alice có được lựa chọn về vật thể mang thông điệp; tệp tin bit nhị phân của đối tượng ba chiều được kết hợp với chuỗi thông điệp ở điểm (2) bằng phép toán XOR hay được biết đến là thuật toán One – Time – Pad (OTP) (Mô phỏng thuật toán: Hình 2.3) [19].



Hình 2.3. Thuật toán One – Time – Pad (OTP)

▪ **Đồng bộ và lưu trữ thông tin:**

- Kết thúc tại điểm (3), Alice có thể đặt vật thể ba chiều tại một mặt phẳng cố định thông qua giao diện của ứng dụng để neo vật thể chứa tin khi Bob có mặt tại điểm đó có thể thấy được vật thể mang thông điệp.
- Để Bob có thể thấy được vật thể thông qua ứng dụng của mình, ngay cả khi Alice không có mặt tại thời điểm đó. Thông tin điểm đặt vật thể và vật thể mang tin phải được lưu trữ trên hệ thống cơ sở dữ liệu; bao gồm hệ thống lưu trữ tệp tin và đồng bộ dữ liệu thời gian thực thông qua cơ sở dữ liệu Firebase.
- Điểm neo được lưu trữ thông qua việc truy xuất API tại Google Anchor Cloud. Trong mô hình mở rộng, hệ thống liên kết với GPS để đặt vật thể tại bất cứ đâu mà người dùng mong muốn.
- Hệ thống máy chủ lưu trữ, truy xuất và cập nhật dữ liệu trong thời gian phiên giao tiếp giữa Alice và Bob có hiệu lực, nhằm duy trì điểm đặt vật thể mang tin. Bao gồm: Cơ sở dữ liệu Firebase (Firebase database), Google Anchor Cloud (ARCore Cloud), GPS (tùy chọn mở rộng),...
- Trong Hình 2.4, mô tả tương quan các thành phần hệ thống máy chủ:
 - + *Firebase database*: Lưu trữ thông tin người dùng. Bao gồm đường dẫn chứa tệp ảnh thông điệp và sinh khóa. Firebase Storage (máy chủ lưu trữ tài nguyên) cung cấp phương thức tạo và quản lý các tệp tin. Ứng dụng và các thành phần khác trong máy chủ sử dụng cơ sở dữ liệu trên Firebase như một nguồn tin cậy.
 - + *Google Anchor Cloud (ARCore Cloud)*: Lưu giữ điểm neo, nói cách khác là vị trí điểm đặt vật thể trong không gian, kết hợp với *hệ thống GPS* mang đến độ chính xác cao tương quan với vị trí của người dùng. Hai thành phần này trao đổi dữ liệu và cập nhật liên tục. Sử dụng nguồn cung từ *Firebase database* để xác định chính xác vật thể cần hiển thị.
 - + Các thành phần này hỗ trợ minh bạch trong bài toán lưu trữ đối tượng trong không gian thực. Mỗi một thành phần mang một tính chất và chức năng riêng; mang tính chuyên biệt hóa, bổ trợ nhau để mục tiêu cuối cùng là hiển thị chính xác đối tượng và thông điệp phía người nhận. Kết thúc quá trình tại điểm (4).



Hình 2.4. Hệ thống máy chủ lưu trữ đối tượng ba chiều trong không gian thực

- *Phát hiện vật thể, giải mã thông tin:*
 - Dữ liệu về điểm neo được thông báo và gửi đến cho Bob khi Bob nhận được thông báo có tin nhắn mới từ phía Alice. Bob được chỉ dẫn tới vị trí đặt mô hình chứa thông điệp của Alice. Bằng thiết bị di động, Bob quét bề mặt xung quanh và tìm ra vị trí của vật thể được cung cấp bởi Google Anchor Cloud. Chỉ dẫn đến vật thể cung cấp bởi GPS API (điểm dừng (5) trong mô hình).
 - Khi ghi nhận được vật thể chứa tin. Cơ sở dữ liệu truy cập tệp tin dữ liệu tương ứng với vật thể đó. Phía máy chủ sẽ:
 - + File dữ liệu gốc của vật thể được lưu một bản sao trên máy chủ lưu trữ. Bản sao này được thực hiện phép XOR bit với tệp tin đã nhúng thông điệp, nhằm thu thập chuỗi kết hợp khóa phiên và thông điệp.
 - + Bản kết hợp được tách ra thành khóa phiên với bản mã. Giải mã khóa phiên dựa trên khóa riêng của Bob thu được khóa bí mật. Khóa bí mật dùng để giải mã thông điệp kèm theo.
 - + Thông điệp hiển thị thông báo trên màn hình ứng dụng. Giúp Bob thu được tin nhắn từ phía Alice.

Dựa trên những phân tích về yêu cầu chức năng cũng như cách thức hoạt động của mô hình, chỉ ra vài nhận xét về ưu và khuyết điểm an toàn của mô hình trên:

- *Ưu điểm:*
 - Chuyên biệt hóa các vùng đảm nhiệm chức năng giúp xử lý tốt thông tin và truy vấn hiệu quả.
 - Dữ liệu che giấu được đảm bảo an toàn đầu – cuối end-to-end vì không những giới hạn về bài toán giải mã mà còn liên quan đến vị trí đặt vật thể trong thế giới thật.
 - Vật thể chứa thông điệp rất khó phân biệt với vật thể nguyên mẫu trước đó vì sự thay đổi các bit mang tính phức tạp phân tích đối với các đối tượng có dung lượng tệp tin lớn. Hơn nữa, nếu không có nguồn dữ liệu mẫu, kẻ tấn công không thể trích xuất được thông tin do không có tệp tin gốc để thực hiện XOR bit.
 - Sự thay đổi các bit, không làm thay đổi kích thước tệp tin nguyên mẫu trước đó. Nếu so sánh về dung lượng trước và sau giấu tin là không đổi.
 - Thay đổi vị trí giấu tin giữa các thành phần kết cấu của đối tượng (đỉnh của vật thể, hướng của bề mặt, màu sắc,...) hoặc giấu tin đan xen, giúp khả năng phát hiện thông tin từ bên ngoài giảm đi một nửa.
- *Khuyết điểm:*
 - Khả năng tiếp cận nội dung không nhanh chóng như những ứng dụng gửi tin nhắn thông thường.
 - Giới hạn các vật thể giấu tin do không có công cụ chuyển đổi định dạng trong thời gian ứng dụng chạy. Định dạng cố định trong khi vật thể ba chiều được hỗ trợ kết xuất bởi nhiều định dạng khác nhau.
 - Do tính chất phiên không duy trì quá lâu, nên khóa phiên có thể bị thay đổi sau một khoảng thời gian. Nếu hệ thống không kết hợp với GPS API thì sẽ không thể lưu giữ vật thể tại vị trí thật của người dùng do một phiên cung cấp bởi Google Anchor Cloud có thể lưu vị trí vật thể chỉ có hơn 24 giờ. Không thể ghi nhận ngày qua ngày.
 - Việc áp dụng các thuật toán mã hóa và giải mã, chưa kể thuật toán giấu tin làm trải nghiệm người dùng bị giảm. Việc xử lý bất đồng bộ với các dữ liệu nhận được từ máy chủ là một điều cần quan tâm.

Từ những nhận định mang tính thử nghiệm của một mô hình giấu tin sử dụng công nghệ AR có tính an toàn cao; ta đi vào phân tích khả năng tích hợp vào ứng dụng thực tế, để thấy được những mặt hạn chế nào còn tồn đọng và những kiến thức nào cần chuẩn bị để xây dựng ứng dụng.

2.3. Chuẩn bị kiến thức, khả năng tích hợp vào ứng dụng

Trước khi tiến hành thực nghiệm trên ứng dụng (nền tảng Android), ta cần hiểu rõ các kiến thức về thành phần hỗ trợ, định dạng giấu tin và các yêu cầu thiết yếu khi triển khai dự án về AR nói chung cũng như phạm vi báo cáo nói riêng. Điều này giúp công việc xây dựng ứng dụng không vượt khỏi khả năng thực hiện đã trình bày.

2.3.1. ARCore và Sceneform

▪ Tổng quan

ARCore là một nền tảng của Google trong việc xây dựng trải nghiệm thực tế ảo tăng cường. Sử dụng nhiều API khác nhau mà nền tảng là kiến thức của OpenGL, một thư viện xử lý đồ họa phổ biến. Tuy nhiên, người phát triển ứng dụng AR, sử dụng ARCore như một công cụ, không cần quan tâm đến kiến thức về OpenGL hay kết xuất vì điều này được hỗ trợ. ARCore giúp thiết bị ghi hình có thể cảm nhận được môi trường xung quanh, hiểu được thế giới xung quanh và tương tác thông tin hiển thị trên đó [20].

Một số nền tảng cho phép xây dựng các ứng dụng, cung cấp API cho trải nghiệm thực tế ảo tăng cường như: Android, IOS, Unity, Unreal,... [20].

ARCore sử dụng 3 từ khóa kỹ thuật (khả năng chính) để tích hợp nội dung chính vào thế giới thực thông qua thiết bị ghi hình:

- *Motion tracking (theo dõi chuyển động)*: Cho phép thiết bị di động có thể hiển thị và theo dõi vị trí tương đối của nó trong thế giới thực.
- *Environmental understanding (thấu hiểu môi trường)*: Cho phép thiết bị di động có thể phát hiện kích thước và vị trí của tất cả các loại bề mặt: Mặt phẳng, mặt đứng và các góc cạnh của bề mặt (ví dụ: Nền đất, mặt bàn hoặc mặt tường,...).
- *Light estimation (tương quan ánh sáng)*: Cho phép thiết bị ghi hình ước lượng điều kiện ánh sáng của môi trường, từ đó điều chỉnh độ sáng và độ bóng của vật thể. Hòa trộn màu sắc nếu ánh sáng màu.

ARCore được thiết kế để có thể hoạt động trên nhiều loại điện thoại Android chạy Android 7.0 (Nougat) trở lên. Đối với một số thiết bị có hiệu số API quá thấp với yêu cầu, vẫn có thể hiển thị hình ảnh vật thể trên màn hình camera, độ trực quan và mức độ trải nghiệm giảm đi đáng kể [20].

▪ Hoạt động

Về cơ bản, ARCore theo dõi vị trí của thiết bị di động khi nó di chuyển và xây dựng những hiểu biết của riêng nó đối với thế giới thật.

Công nghệ theo dõi chuyển động của ARCore sử dụng thiết bị ghi hình (camera của thiết bị di động) để xác nhận những điểm đặc trưng và theo dõi cách chuyển động của các điểm đó theo thời gian. Kết hợp chuyển động các điểm và đọc chúng từ cảm biến quán tính của thiết bị di động. ARCore xác định cả vị trí và hướng của thiết bị di động thông qua chuyển động của chúng trong không gian [20].

Ngoài việc xác định các điểm như đặc trưng quan trọng. ARCore có thể phát hiện bề mặt phẳng (ví dụ: Nền nhà, mặt bàn, mái nhà,...) và có thể ước lượng ánh sáng trung bình xung quanh khu vực đó. Kết hợp khả năng theo dõi chuyển động và định lượng ánh sáng giúp ARCore có thể xây dựng những hiểu biết riêng của nó đối với thế giới thực [20].

Những hiểu biết của ARCore đối với thế giới thực giúp ta có thể đặt vật thể đa chiều trong không gian, chú thích hoặc mang những thông tin khác tích hợp hoàn hảo với thế giới thực. Motion tracking có nghĩa là ta có thể di chuyển xung quanh những vật thể đó và quan sát nó từ mọi góc độ. Thậm chí sau khi rời khỏi vị trí khuất tầm nhìn vật thể và quay trở lại. nó vẫn ở đó [20].

▪ **Giới thiệu về Sceneform**

Sceneform là một framework 3D được giới thiệu bởi Google trong sự kiện IO'. Những API được cung cấp bởi Sceneform giúp phát triển các ứng dụng ARCore trở nên dễ dàng và nhanh hơn do hầu hết các khái niệm cơ bản đều đã được xử lý nội bộ, như phát hiện về mặt và cân bằng sáng, tất cả đều đã được xử lý bởi Sceneform mà không cần tìm hiểu về đồ họa 3D hoặc OpenGL [20].

Một số API chính thường được sử dụng nhất trong ứng dụng:

- *ArFragment*: Thành phần được thêm vào bố cục layout của ứng dụng. Đóng vai trò như một khung quan sát chính. Nó tự động kiểm tra yêu cầu phiên bản ARCore trong thiết bị và hỏi người dùng về quyền truy cập camera trong thời gian thực thi lần đầu.
- *ArSceneView*: Nó kết xuất các hình ảnh camera và ánh sáng khi bề mặt phẳng (*Planes*) được phát hiện bởi ARCore. Ta có thể sử dụng *ArSceneView* trực tiếp trong ứng dụng tuy nhiên trong trường hợp này cần đảm bảo rằng phải có phương thức kiểm tra thực thi trong thời gian chạy và kiểm tra quyền truy cập phần cứng thiết bị.
- *ModelRenderable*: Cho phép ta tải đối tượng 3D từ đường dẫn đích. Và trả về đối tượng *Renderable* có thể được kết xuất ra màn hình.
- *Node*: Là vật thể ảo cần được kết xuất. Nó chứa tất cả những thông tin Sceneform cần để kết xuất đối tượng. Ta có thể tương tác với các đối tượng

được kết xuất tại Node như: Transform (biến đổi), chuyển động (animate) và rotate (xoay) các Node.

▪ Các định dạng hỗ trợ

Sceneform và ARCore hỗ trợ tài nguyên 3D theo những định dạng sau:

- *.OBJ: Là định dạng hình ảnh 3D tiêu chuẩn có thể được xuất và mở bằng các chương trình chỉnh sửa hình ảnh 3D khác nhau. Chứa các đối tượng ba chiều bao gồm tọa độ, bản đồ kết cấu, mặt đa giác và các thông tin khác. Có thể tham chiếu đến nhiều tệp *MTF* chứa vật liệu tạo chi tiết cho bề mặt. Có những đặc điểm:
 - + Hỗ trợ kết xuất và review hình ảnh trong Android Studio.
 - + Định dạng được hỗ trợ hầu hết ở các website cung cấp nguồn tài nguyên
 - + Đối tượng cần có khâu tái kết cấu thành 2 file định dạng là *SFB* và *SFA* mới có thể sử dụng như một tài nguyên cho AR.
 - + Không có chuyển động.
 - + Không thể kết xuất trong thời gian chạy ứng dụng.
- *.FBX: Là định dạng được sử dụng để trao đổi dữ liệu hình học và chuyển động. Ta có thể sử dụng 2 chương trình khác nhau để hiệu chỉnh và cho ra độ trung thực cao. Các tệp này thường được sử dụng trong phát triển phim, trò chơi và thực tế ảo (AR/VR). Có các đặc điểm:
 - + Hỗ trợ chuyển động.
 - + Cần nạp tài nguyên trước khi sử dụng.
 - + Không kết xuất trong thời gian chạy ứng dụng.
- *.glTF và *.GLB: Là file chứa mô hình 3D, lưu trữ các mô tả của đối tượng dưới dạng *JSON*, bao gồm: Các nút phân cấp, góc máy, vật liệu,... [21]. Ngoài ra nó còn chứa thông tin về các lưới và chuyển động. Tuy nhiên trong Sceneform, *glTF* không được hỗ trợ chuyển động. Có đặc điểm sau:
 - + Tham chiếu đến tệp nhị phân *.BIN chứa dữ liệu hình học, chuyển động, màu sắc,... Được đọc từ định dạng *glTF* (bắt buộc).
 - + Tệp Shader (Trình đồ bóng *.GLSL) chứa trình đồ bóng (không thường thấy ở các đối tượng 3D).
 - + Các file hình ảnh (*.JPG, *.PNG, ...) chứa kết cấu cho mô hình 3D, các chi tiết trên bề mặt đối tượng (có hoặc không).
 - + Không cần phải khai báo, nạp tài nguyên trước khi sử dụng.
 - + Có thể kết xuất trong thời gian ứng dụng chạy.

- **ARCore và nền tảng Android**

Tuy ARCore cũng tích hợp cho nền tảng IOS nhưng cách đó không lâu vào khoảng Tháng 6 năm 2017. IOS cho ra đời ARKit là tiêu chuẩn riêng cho thiết kế ứng dụng AR cho nền tảng IOS.

Trước đó, Vuforia cũng được cấp giấy phép vào năm 2015 là một trong những đối thủ cạnh tranh sớm của ARCore. Xét về nhiều mặt thì vẫn chưa biết được hẳn là nền tảng AR nào là ưu điểm hơn. Song ARCore được biết đến rộng rãi, cải thiện và công bố nhiều tính năng mỗi năm. Vì thế, phát triển ứng dụng AR trên nền tảng di động nói chung và nền tảng Android nói riêng không thể không nhắc đến đề cử ARCore.

Hơn thế nữa, việc xây dựng ứng dụng ARCore đã được hỗ trợ nhiều hơn trước. Từ việc tìm hiểu các học thuyết về đồ họa ba chiều đến việc lập trình C# với thư viện OpenGL để xử lý các sự kiện logic liên quan đến đồ họa nay đã được gói gọn trong một nền tảng. Việc giới thiệu Sceneform hỗ trợ người dùng lược bỏ đi những thao tác mặc định thủ công của một ứng dụng AR như: Kiểm tra quyền truy cập phần cứng, phát hiện bề mặt đặt vật thể, định lượng ánh sáng xung quanh,... đã giảm thiểu đi nhiều thời gian cấu hình ứng dụng. Giúp việc xây dựng ứng dụng AR trở nên dễ dàng và nhanh chóng.

2.3.2. Định dạng glTF trong giấu tin trong với ứng dụng AR

- **Giới thiệu**

Theo KhronosGroup – tập đoàn sáng lập ra định dạng glTF. glTF Transmission Format (glTF) là một API trung lập về định dạng phân phối tài nguyên tại thời điểm thực thi (runtime). glTF làm cầu nối giữa công cụ sáng tạo nội dung 3D và ứng dụng mô hình 3D bằng cách cung cấp một định dạng hiệu quả, có thể mở rộng, có thể tương tác để truyền tải nội dung 3D [21].

- **Nguyên nhân ra đời**

Các loại định dạng 3D tiêu chuẩn công nghệ hiện đại cho phép chia sẻ tài nguyên giữa các công cụ xây dựng đối tượng 3D khác nhau. Tuy nhiên các loại định dạng này đều không được tối ưu hóa để cho tốc độ tải về và tốc độ kết xuất nhanh trong thời gian thực thi chương trình.

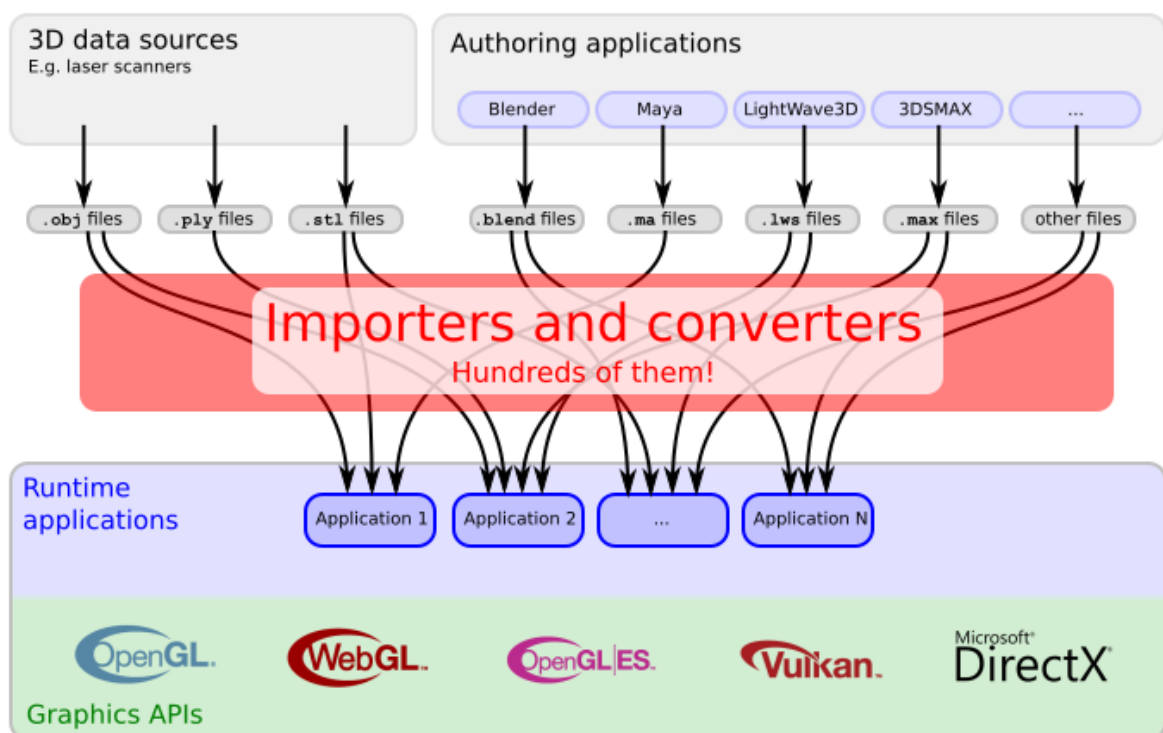
Các tệp có xu hướng phát triển rất lớn và các ứng dụng cần thực hiện xử lý đáng kể để tải các tài nguyên đó vào trình xử lý trên GPU.

Các ứng dụng có hiệu suất cao (về độ quy mô của đối tượng, kích thước của đối tượng, vật liệu bên ngoài đối tượng, kết cấu bên trong,...) hiếm khi tải các định

dạng mô hình trực tiếp. Thay vào đó, các mô hình này được xử lý ngoại tuyến, nạp theo một định dạng tương ứng theo các quy ước ứng dụng riêng để tối ưu hóa khả năng kết xuất trong thời gian ứng dụng thực thi.

Sự ra đời của điện toán ba chiều dựa trên các thiết bị di động và web, các lớp ứng dụng mới đã xuất hiện đòi hỏi tốc độ tải phải nhanh, nhất là đối với các tài sản ba chiều được tiêu chuẩn hóa.

Trong Hình 2.5, thể hiện hệ thống đường ống nội dung 3D (3D content pipeline) là một mô hình đa công cụ xây dựng và quản lý các đối tượng ba chiều trong cùng một hệ thống ứng dụng. Dưới sự tác động của nhiều công cụ chế tác và hiệu chỉnh nội dung của đối tượng ba chiều thì việc kết xuất ra cùng một định dạng là một bài toán khó khăn; điều này đòi hỏi phải có nhiều thao tác chuyển đổi đầu-cuối để xuất ra tiêu chuẩn định dạng khác nhau; phục vụ cho hệ thống ứng dụng hiển thị đối tượng trong thời gian thực thi (runtime).

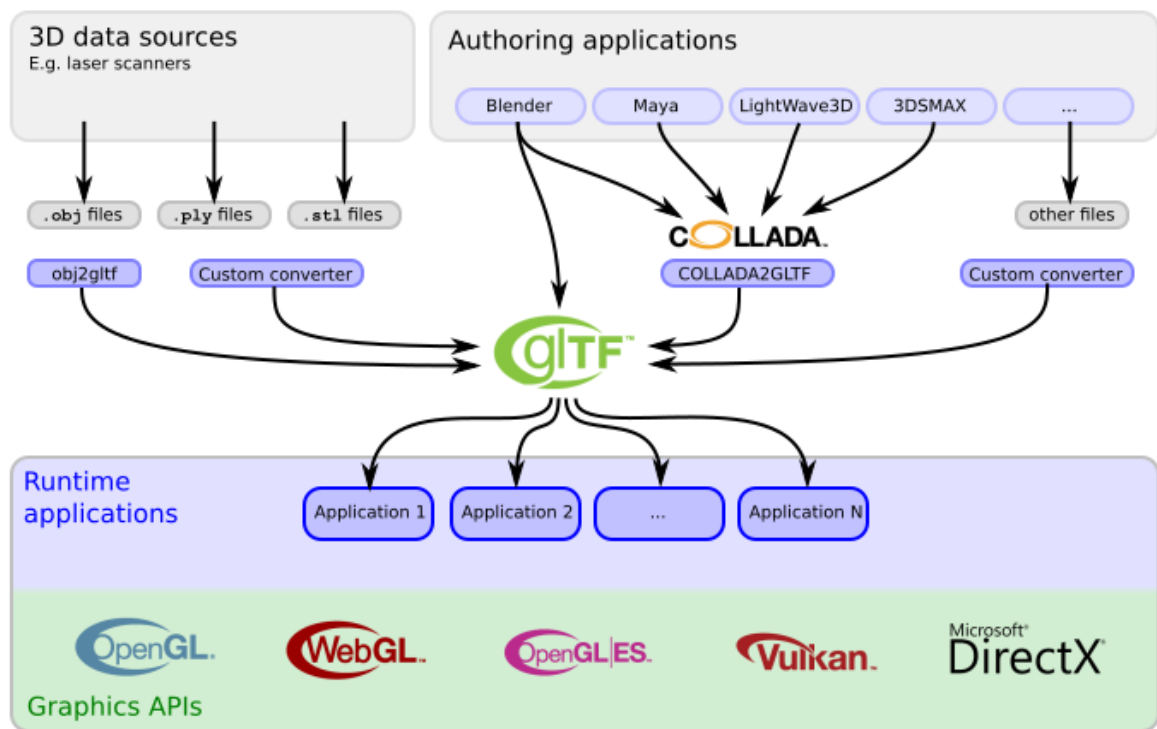


Hình 2.5. Cấu trúc đường ống (pipeline) nội dung 3D trước đây

glTF giải quyết các vấn đề này bằng cách cung cấp định dạng tập trung cho các nhà phát triển và các đối tượng có thể kết xuất trong thời gian chương trình được nạp (Hình 2.6), nghĩa là trong thời gian chưa trình được thực thi, đối tượng ba chiều được nạp vào bộ nhớ chương trình và sẵn sàng hiển thị trong môi trường ứng dụng cho người xem với thời gian xử lý tối thiểu nhất.

Do định dạng này kết hợp với mô tả trong JSON nên cú pháp dễ phân tích hơn nhiều với một hoặc nhiều tệp nhị phân đại diện cho hình học, hình động và các kết cấu hay chất liệu bề mặt của đối tượng 3D.

Dữ liệu nhị phân có thể lưu trữ theo cách có thể được tải trực tiếp vào bộ đệm GPU mà không cần phân tích thêm hoặc thao tác khác. Sử dụng phương pháp này glTF có thể bảo toàn các cảnh phân cấp với các nodes (nút), meshes (lưới), hệ thống camera, hình động và vật liệu. Đồng thời cho phép phân phối các bit linh hoạt và hiệu quả [21].



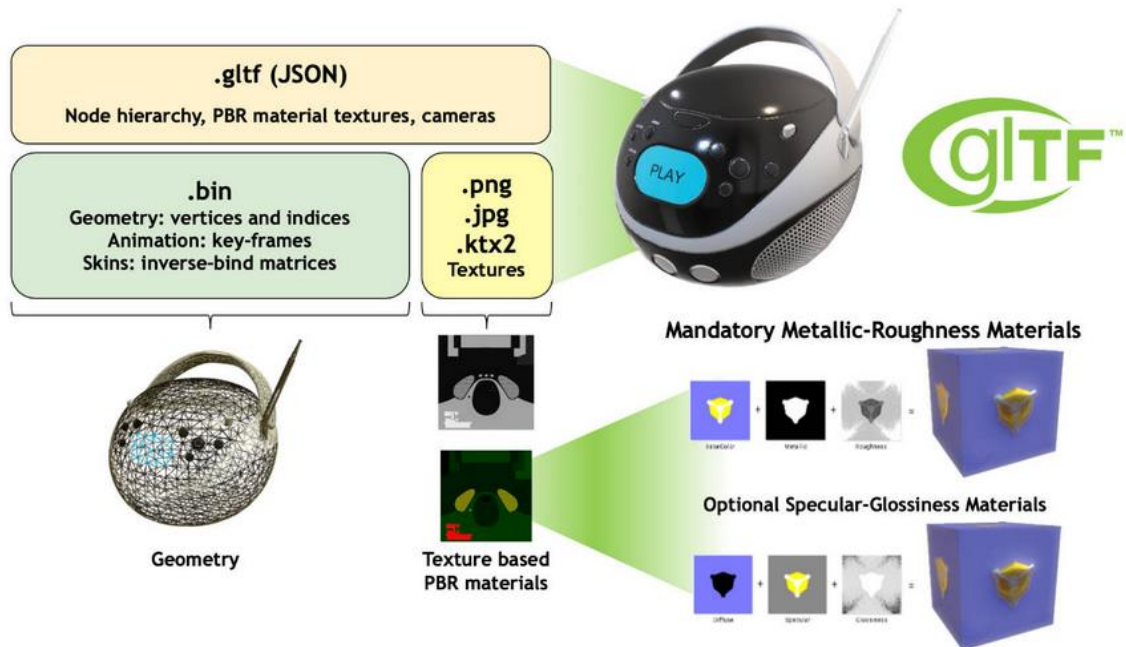
Hình 2.6. Cấu trúc đường ống nội dung 3D kết hợp glTF

▪ Thành phần cơ bản:

Trong Hình 2.7 mô tả cấu trúc cơ bản của định dạng glTF. Tài nguyên glTF là các tệp JSON kết hợp với việc hỗ trợ dữ liệu ngoài. Cụ thể, tài nguyên glTF được đại diện bởi các thành phần sau:

- Một tệp tin có định dạng JSON (.gltf) chứa tất cả các mô tả phân cảnh của đối tượng bao gồm: Phân cấp điểm có trên đối tượng (node hierarchy), vật liệu (materials), các kết cấu chuyển động cameras như: Lưới (meshes), chuyển động (animation) và kết cấu khác.
- Tệp dữ liệu nhị phân (.bin) bao gồm cấu trúc hình học (các đỉnh của bề mặt và các chỉ số đại diện của chúng), dữ liệu chuyển động (các khung diễn hoạt của đối tượng) và các phần đệm.

- Các tệp tin hình ảnh (.jpg, .png, .ktx2) cho các chi tiết hiển thị trên các bề mặt của đối tượng. Các đối tượng có chi tiết được quy định thêm hai thuộc tính là: Metallic-Roughness Material (độ nhám, kim loại cho đối tượng kim loại, phi kim) và Specular-Glossiness Material (độ bóng cho các đối tượng có bề mặt phản chiếu).



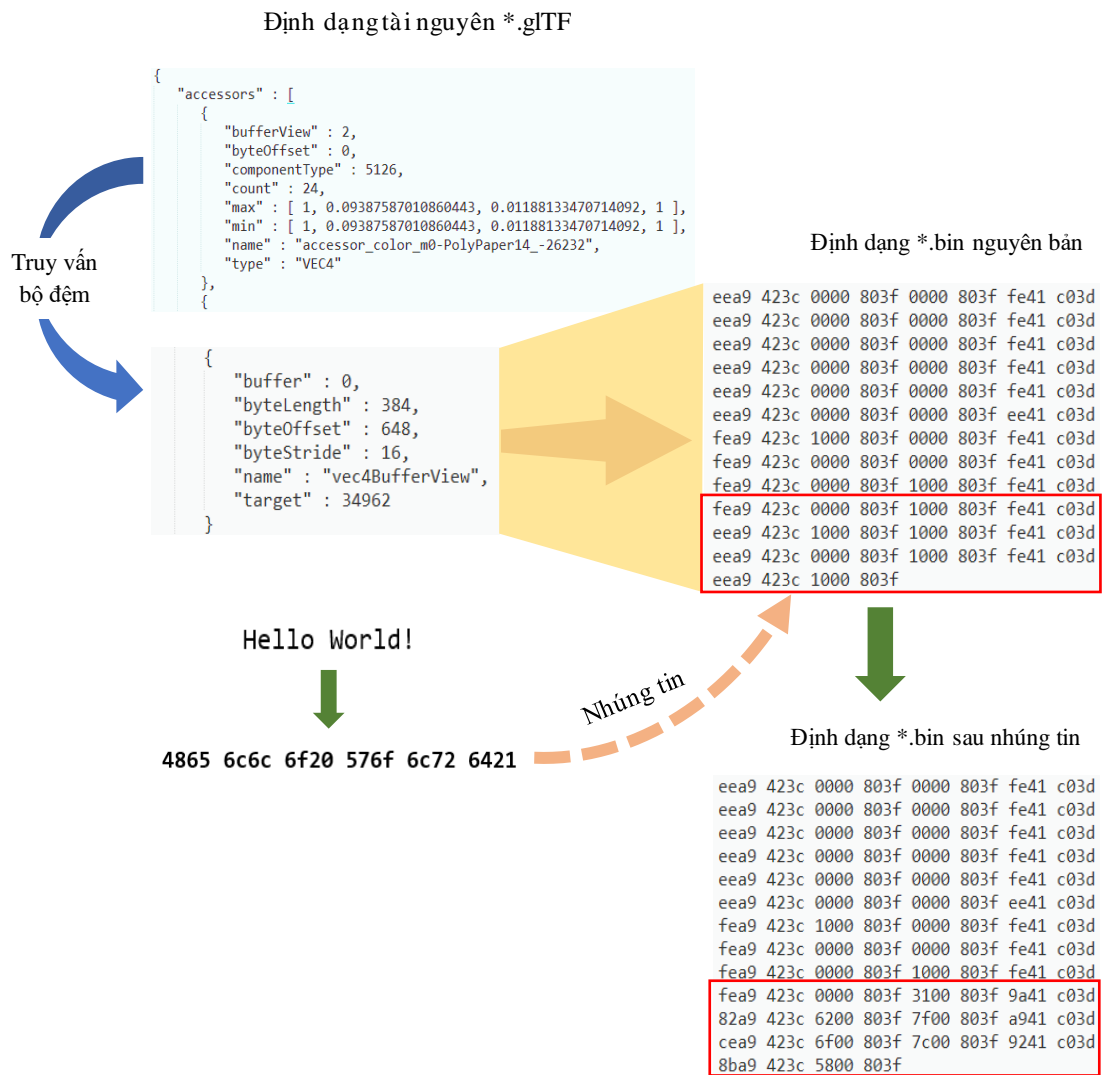
Hình 2.7. Mô tả về cấu trúc của định dạng glTF

▪ Định dạng glTF và tương quan giấu tin

Thông thường, trong ứng dụng Android việc xử lý các dữ liệu JSON là điều thường xuyên do đa số các API trả về dữ liệu dưới dạng JSON cho thấy sự tiện lợi trong quá trình xử lý và đọc hiểu nội dung đối tượng mà không cần bất kỳ sự trợ giúp từ phần mềm thứ ba nào [21].

Định dạng glTF cơ bản nhất bao gồm một tệp tin kiến trúc đối tượng và tệp tin nhị phân:

- Đối với tệp tin cấu trúc (.glTF) ta có trích xuất lấy thông tin các thành phần riêng lẻ của đối tượng, lưu trữ thành các biến riêng lẻ. Từ đó phân tích được quy mô, tiềm năng giấu thông tin của đối tượng hiện tại (Hình 2.8).
- Đối với tệp tin nhị phân (.bin) lưu trữ dữ liệu dưới dạng bit. Nếu ở trên ta đã có cấu trúc của đối tượng thì khi ánh xạ qua tệp tin nhị phân, ta có thể thu thập chính xác các thành phần của đối tượng.
- Do tính chất thông tin được lưu dưới dạng nhị phân nên khi chuyển thông điệp từ dạng chữ sang nhị phân, ta dễ dàng ẩn giấu chúng cùng với tệp tin tương ứng (Hình 2.8).



Hình 2.8. Áp dụng nhúng tin vào cấu trúc glTF

- Ưu điểm của định dạng glTF/GLB trong ứng dụng AR là có khả năng kết xuất hình ảnh trong thời gian ứng dụng khởi chạy. Trong khi đó các định dạng khác phải trực chuyển sang một định dạng thay thế trước khi ứng dụng khởi chạy. Điều này khiến ta không thể lấy được bất kỳ thông tin nào liên quan đến kết cấu của đối tượng 3D, chỉ có thể tương tác sự kiện đối với chuyển động của vật thể.
- Do có khả năng kết xuất đối tượng 3D trong quá trình ứng dụng chạy, giúp cho ta có thể thay đổi các đối tượng đang truy cập trong thời gian thực. Đồng thời có thể xử lý được đối tượng như: Thay đổi vị trí, thay đổi các đỉnh – kết cấu của đối tượng, thay đổi màu sắc,... mà không gây ra lỗi kết xuất đối tượng.

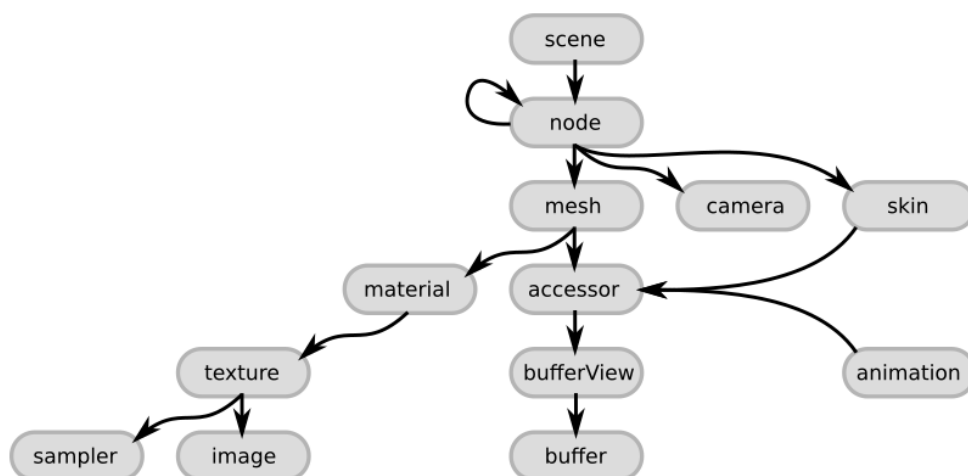
- Điểm sáng trong vận dụng định dạng glTF là việc *thay đổi kết cấu của đối tượng, không làm thay đổi kích thước của tệp tin*, là một ưu điểm mà glTF được lựa chọn trở thành môi trường chứa thông tin mật.

▪ Cấu trúc tệp tin glTF

Để thuận lợi trong việc đọc, truy xuất và chỉnh sửa cấu trúc của đối tượng ba chiều, việc tìm hiểu cấu trúc xây dựng tệp tin là một điều quan trọng:

Tệp tin glTF được xây dựng dựa trên cấu trúc file JSON. Các khung yếu tố cấu thành nên đối tượng được lưu trữ dưới dạng mảng; chúng có thể được truy cập bằng việc sử dụng định danh và chỉ số tương ứng của đối tượng trong mảng [21].

Các định dạng này cũng thường được dùng để định nghĩa các mối quan hệ giữa các đối tượng. Ví dụ: Một đối tượng có nhiều mảng lưới, trong mỗi lưới có nhiều nút khác nhau (hay còn gọi là đỉnh) chúng có tương quan với nhau và ánh xạ qua lại trong một tệp tin.



Hình 2.9. Cấu trúc cây JSON trong glTF

Hình 2.9 minh họa cấu trúc cây JSON trong tệp tin *.glTF. Theo như mô tả ở trên, đối tượng 3D có nhiều nhánh thuộc tính, liên kết với nhau bằng thao tác gọi khi khai báo thuộc tính con cho một nhánh cha. Các yếu tố này được nêu ra nhằm đưa ra các nhìn tổng quan và liên kết giữa các thành phần trong đặc tả glTF. Giúp giải thích chi tiết các mối quan hệ giữa các thành tố cấu thành nên đối tượng:

- *Scene (phân cảnh)*: Là điểm vào của phân cảnh được lưu trữ trong glTF. Nó đề cập đến các nút được xác định trong phân cảnh (ví dụ: Nút bắt đầu, nút kết thúc... đối tượng).
- *Node (nút – đỉnh)*: Là một nút trong hệ thống phân cấp phân cảnh. Nó có thể chứa biến đổi được phép có của đối tượng (ví dụ: Xoay, dịch chuyển,

thu phóng,...). Có thể đề cập đến các nút con, có khi là lưới, vật liệu, camera,...

- *Camera*: Định nghĩa cấu hình khung nhìn cho kết xuất phân cảnh.
- *Mesh (lưới)*: Mô tả cách mà một đối tượng hình học xuất hiện trong phân cảnh. Thành phần này thường được sử dụng để truy cập dữ liệu hình học và định nghĩa material (chi tiết, hoa văn) khi đối tượng được kết xuất.
- *Skin (chi tiết, họa tiết, hoa văn)*: Định nghĩa tham số cho hoạt tiết được yêu cầu bởi đỉnh.
- *Material (nguyên liệu)*: Bao gồm các tham số định nghĩa sự xuất hiện của một đối tượng. Nó sử dụng đối tượng kết cấu (texture) cho bề mặt của lưới đối với các đối tượng được kết xuất hình học.
- *Texture (kết cấu)*: Định nghĩa là một sampler (thành phần lấy mẫu) và một hình ảnh. Thành phần lấy mẫu định nghĩa nên làm thế nào để đặt hình ảnh vào đối tượng.

Một đối tượng theo định dạng glTF có khá nhiều thành phần chi phối việc hiển thị đối tượng. Trong đó, tệp tin JSON đóng vai trò đưa ra cấu trúc phân cảnh chung, liên kết các nhánh thuộc tính con (như: Nút, camera, chuyển động). Thêm vào đó, nó bao gồm đường dẫn ngoài trỏ đến thư mục nhị phân và các hình ảnh kết cấu họa tiết kèm theo.

▪ Liên kết dữ liệu giữa glTF và bộ đệm trong tệp tin binary

Để bắt đầu đọc và phân tích đối tượng có định dạng glTF là việc bắt đầu phân tích cấu trúc JSON. Sau khi cấu trúc được phân tích, các đối tượng bộ đệm (tệp tin nhị phân hay còn gọi là buffer) và các tệp tin chi tiết bề mặt (images) được gọi ra tương ứng với các mảng hình ảnh được truy xuất [21].

Mỗi đối tượng này có thể được gọi ra bởi một khối dữ liệu nhị phân. Thông thường, dữ liệu được lưu trong một mảng để chúng có thể được tra cứu bằng các sử dụng cùng một chỉ mục để tham chiếu đến bộ đệm hoặc đối tượng hình ảnh thuộc về chúng [21].

Dữ liệu nhị phân này chỉ là một khối bộ nhớ thô được đọc từ URI của bộ đệm; không có ý nghĩa hoặc cấu trúc vốn có. Lưu trữ dữ liệu ở dạng nhị phân cho phép nó được truyền qua web hiệu quả hơn so với định dạng JSON và dữ liệu nhị phân có thể truyền trực tiếp đến trình kết xuất mà không cần giải mã hoặc xử lý trước [21].

Trong tệp tin bin, ta cần tìm hiểu rõ phạm vi và quy ước lưu trữ của các khối nhị phân. Tuy không tuân theo bất kỳ quy ước lưu trữ nào như bằng việc ánh xạ các thuộc tính cha vào thuộc tính con và truy xuất đến vùng nhớ đệm nhị phân để gọi dữ

liệu đã giúp giảm lược đi phần nào độ phức tạp xử lý dữ liệu trên các đối tượng 3D. Thuận lợi cho bài toán ẩn giấu thông tin sau này. Một định dạng glTF cơ bản không thể thiếu một trong ba thành phần quan trọng để cấu thành nên đối tượng và móc nối các dữ liệu nhị phân của tệp tin binary là Buffers, Buffer Views và Accessors [21]:

- *Buffers (bộ đệm)*: Đại diện cho một khối dữ liệu nhị phân thô (raw binary data), vốn không có cấu trúc hay ý nghĩa nào. Dữ liệu này được thể hiện bởi việc dùng bộ đệm của URI. URI này có thể trỏ đến một tệp bên ngoài hoặc là một nguồn dữ liệu mã hóa thành nhị phân trực tiếp trong tệp JSON. Các phần dữ liệu của buffer (bộ đệm) có thể được chuyển đến trình kết xuất dưới dạng các thuộc tính đỉnh, chỉ mục chứa các thông tin về chi tiết / họa tiết bề mặt hoặc chuyển động của đối tượng 3D.
- *BufferViews (bộ đệm khung nhìn)*: Trong bước đầu phân tích cấu trúc dữ liệu từ một bộ đệm (buffer) nêu trên là truy vấn đến bufferView. Một bufferView đại diện cho một phần dữ liệu của buffer, nó được chia nhỏ tùy thuộc vào đối tượng đó có nhiều hay ít thuộc tính. Trong một phần của bufferView nó định nghĩa việc sử dụng số lượng các bit nhị phân trong tệp tin binary như: Vị trí xuất hiện thuộc tính mới, chiều dài, kiểu bytes... Với tổng số bufferView ta biết được buffer chiếm được bao nhiêu thành phần trong tổng số nội dung tệp tin binary.
- *Accessors (bộ truy cập)*: Một đối tượng accessor thường đề cập đến một bufferView và chứa giá trị định nghĩa loại và bố trí dữ liệu của bufferView đó. Kiểu dữ liệu trong bộ truy cập được mã hóa bởi thuộc tính phân loại (type) và kiểu thành phần (componentType). Các giá trị này là một chuỗi phân biệt các dữ liệu thuộc: vô hướng (SCALARS), có hướng (VECTORS) hay ma trận (MATRICES):
 - + *Scalars*: Cho ra kết quả là giá trị vô hướng.
 - + *Vectors*: Cho kết quả là giá trị vector có hướng ba chiều.
 - + *Matrices*: Cho kết quả là ma trận có kích thước 4x4

Kiểu dữ liệu thành phần bao gồm hai giá trị:

- + 5126: Kiểu FLOAT đại diện cho các thuộc tính có giá trị chênh lệch theo định dạng số thực, có dấu chấm động.
- + 5123: Kiểu UNSIGNED_SHORT đại diện cho các thuộc tính có giá trị số nguyên không dấu.

2.4. Kết luận chương 2

Chương này đã chỉ ra yêu cầu cơ bản trong xây dựng một hệ thống ứng dụng thực tế ảo. Từ việc xem xét các yêu cầu cần thiết và khó khăn trong quá trình triển khai, trong phần này đã trình bày được mô hình giấu tin trong hệ thống thực tế ảo tăng cường (AR).

Một hệ thống truyền và xử lý thông điệp trên nền tảng thực tế ảo bao gồm các quá trình: Ghi nhận dữ liệu, chuyển đổi thông tin, đồng bộ và lưu trữ, cuối cùng là phát hiện và giải mã thông điệp. Hệ thống máy chủ bao gồm: Hệ thống lưu trữ cơ sở dữ liệu thời gian thực Firebase, hệ thống chia sẻ điểm neo vật thể Google Anchor Cloud và thành phần hỗ trợ định vị GPS API. Sử dụng một số thuật toán giấu tin như: Kết hợp mã hóa khóa đối xứng và bất đối xứng, sử dụng thuật toán One Time Pad (OTP). Từ những yếu tố cấu thành trên, đảm bảo cho thông tin khó bị phát hiện, mức độ khép kín thông tin cao và bảo mật đến tay người dùng cuối.

Xác định nền tảng sử dụng là AR/Sceneform kèm theo đó là định dạng tệp tin glTF với đặc trưng kết xuất đối tượng ba chiều trong thời gian ứng dụng thực thi. Đây là hướng sáng trong việc tìm kiếm khả năng giấu tin trong các ứng dụng thực tế ảo nói chung và thực tế ảo tăng cường (AR) nói riêng.

Trong định dạng file glTF, các thuộc tính đối tượng được phân cấp rõ ràng. Ưu điểm của việc giấu tin trong các bit ít quan trọng là không làm thay đổi dung lượng file chứa, đối tượng không bị thay đổi kết cấu sau khi thông điệp được nhúng vào.

CHƯƠNG 3. TRIỂN KHAI ỨNG DỤNG MINH HỌA GIẤU TIN TRONG MÔI TRƯỜNG THỰC TẾ ẢO

3.1. Yêu cầu đối với ứng dụng

Trong phần 2.2 và phần kết luận 2.4 ta đã nêu ra được tổng quan quy trình thu thập và ẩn giấu thông điệp từ người dùng. Việc phân tích và xây dựng ứng dụng phải chỉ ra được những phương thức, mối liên hệ trong chương trình xây dựng tương ứng với từng bước trong quy trình đã nêu.

Điểm quan trọng trong việc giấu tin là làm rõ mối quan hệ giữa thông điệp của người dùng và định dạng glTF đã trình bày ở phần 2.3. Minh chứng được ưu điểm của định dạng glTF là có thể kết xuất dữ liệu đối tượng trong thời điểm ứng dụng thực thi – điều không thể thực hiện đối với các định dạng tiêu chuẩn.

Xây dựng được phương pháp chuyển đổi giữa giấu tin vào các bit trọng yếu và thứ yếu để thấy được sự khác nhau về khả năng ảnh hưởng đối với thông tin đã thu thập từ người dùng.

Mô phỏng phương pháp đặt vật thể trong thế giới thật, kiểm tra sự kiện. Lưu trữ và đồng bộ hóa được dữ liệu người dùng lên máy chủ. Lắng nghe thay đổi khi người dùng nhập một thông điệp mới.

Bước đầu áp dụng thuật toán mã hóa vào thông điệp và triển khai hệ thống chia sẻ vị trí trong thế giới thật.

3.2. Phát biểu bài toán và xây dựng kịch bản

3.2.1. Phát biểu bài toán

- *Giai đoạn 1: Phân tích định dạng và nhúng thông điệp vào đối tượng 3D*

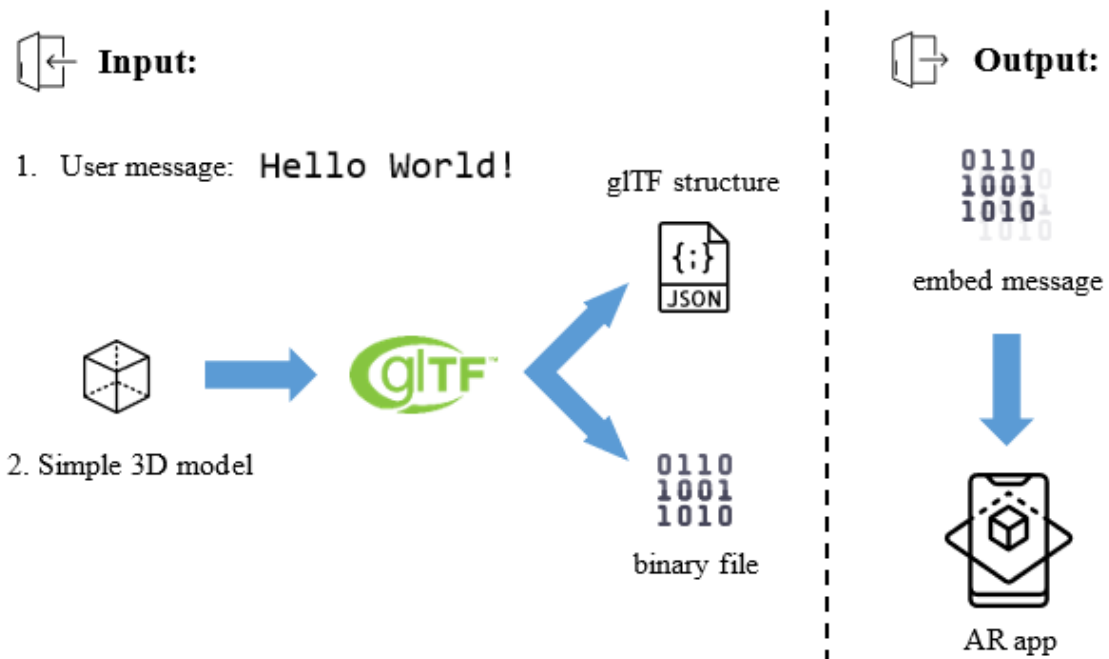
Với một thông điệp đầu vào của người dùng, bằng cách đọc cấu trúc của định dạng đưa thông điệp vào bộ đệm của hệ thống định dạng glTF. Sau đó, sử dụng AR để kết xuất đối tượng 3D ra ngoài giao diện người dùng (Hình 3.1).

- *Giai đoạn 2: Cập nhật cơ sở dữ liệu và đặt vật thể trong không gian thực*

Đối tượng 3D mặc định được lưu trữ trong thư mục tài nguyên (assets) trong hệ thống Android. Ngay cả khi không có bất cứ đối tượng 3D nào nhận từ người dùng hoặc hệ thống truy xuất tài nguyên không sẵn sàng (bận/mất kết nối) thì luôn có đối tượng dự phòng để duy trì trải nghiệm người dùng.

Do tính chất không thể hiệu chỉnh nên tệp tin không thể chỉnh sửa trong thư mục assets. Kèm theo việc đối tượng mang thông tin là tệp tin nhị phân và phát

triển mô hình chia sẻ vật thể 3D do vậy nên vật mang tin cần được lưu trữ trên một hệ thống dữ liệu tập trung.



Hình 3.1. Giai đoạn phân tích định dạng và ẩn giấu thông điệp

▪ *Giai đoạn 3: Trích rút thông điệp gốc và áp dụng mật mã*

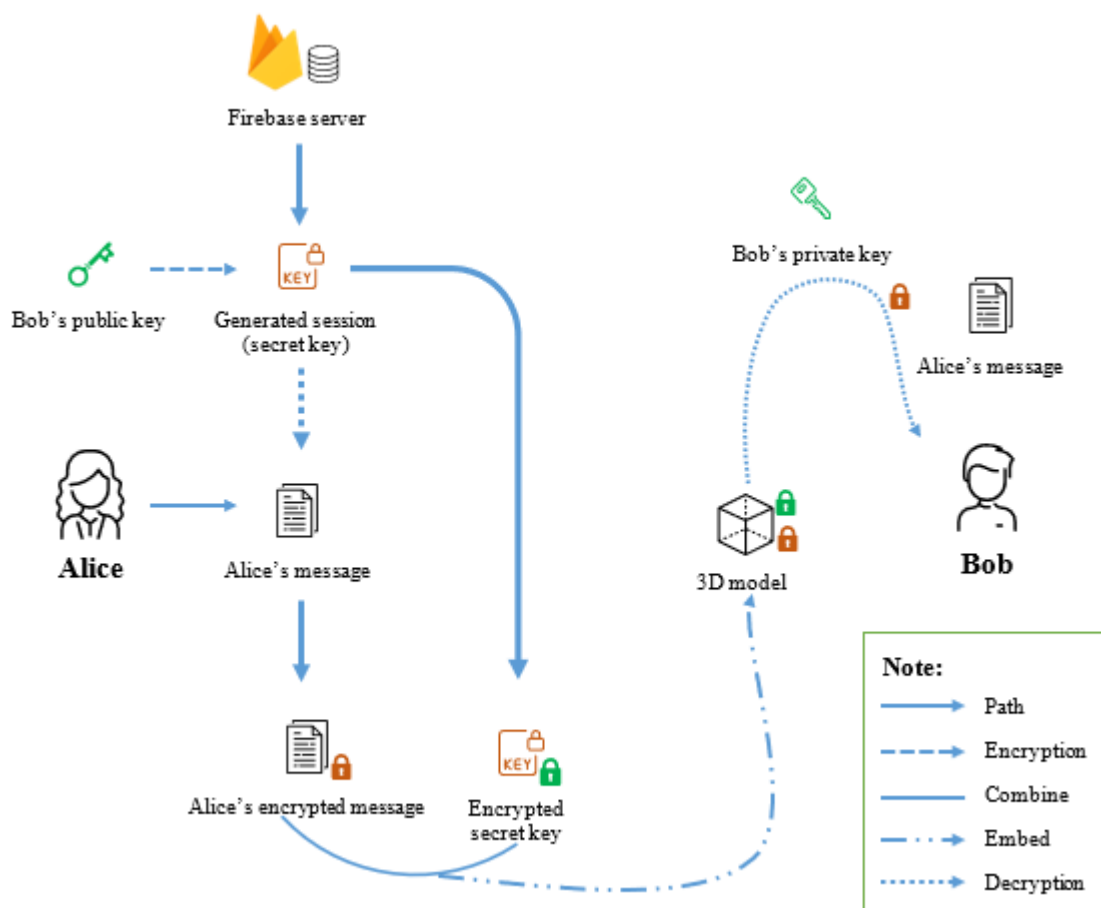
Sau khi hoàn tất các yêu cầu ở hai giai đoạn trên. Để đảm bảo tính bí mật và tính không thể tiếp cận thông điệp đối với kẻ tấn công. Thực hiện kết hợp mô hình mật mã, giữa khóa phiên và khóa bất đối xứng qua mỗi lần thực hiện trao đổi giữa người dùng với nhau (đã trình bày trong mục 2.2).

Việc trích rút thông điệp gốc tương ứng với quá trình giải mã. Thực hiện phân tích lại cấu trúc JSON của định dạng glTF ta tìm ra các miền thuộc tính của đối tượng 3D. Định tính chế độ nhúng tin để lấy chính xác vùng mang thông tin được giấu trong tập tin nhị phân. Từ đó giải mã và thu được thông điệp gốc (Hình 3.2).

▪ *Giai đoạn 4: Chia sẻ vật thể qua đám mây và dịch vụ GPS*

Hoàn tất yêu cầu sử dụng mật mã trong ẩn giấu thông tin. Các khóa được lưu trữ trong thiết bị người dùng/hay trên hệ thống máy chủ (tùy chọn cấu hình). Việc này trợ giúp cho việc chia sẻ vật thể qua đám mây và lưu trữ các thông tin bao gồm: vị trí đặt vật thể, vị trí của người dùng,... Triển khai hệ thống Google Anchor Cloud và GPS API (Hình 2.4).

Kết quả tổng quan: Người dùng nhận được thông tin gửi cho nhau qua ứng dụng bằng cách quét vị trí vật thể 3D được đặt trước đó.



Hình 3.2. Mô hình mật mã giấu tin áp dụng trong ứng dụng AR

3.2.2. Xây dựng kịch bản mô phỏng

Do giới hạn báo cáo nên mức độ thử nghiệm thay đổi từ đa mục tiêu sau khi hoàn thành cả 4 giai đoạn triển khai thành hai giai đoạn với đơn mục tiêu người dùng. Mô hình triển khai được mô tả như sau:

Theo minh họa (Hình 3.3), tại điểm (1) người dùng gửi thông điệp đến ứng dụng. Tại thiết bị của người dùng, ứng dụng xử lý dữ liệu thô của người dùng thông qua phép biến đổi để từ thông điệp dạng chữ sang dạng nhị phân (được trình bày dưới hiển thị thập lục phân – hexa để rút gọn phương pháp XOR bit chuyển thành XOR các kí tự thập lục phân) [19].

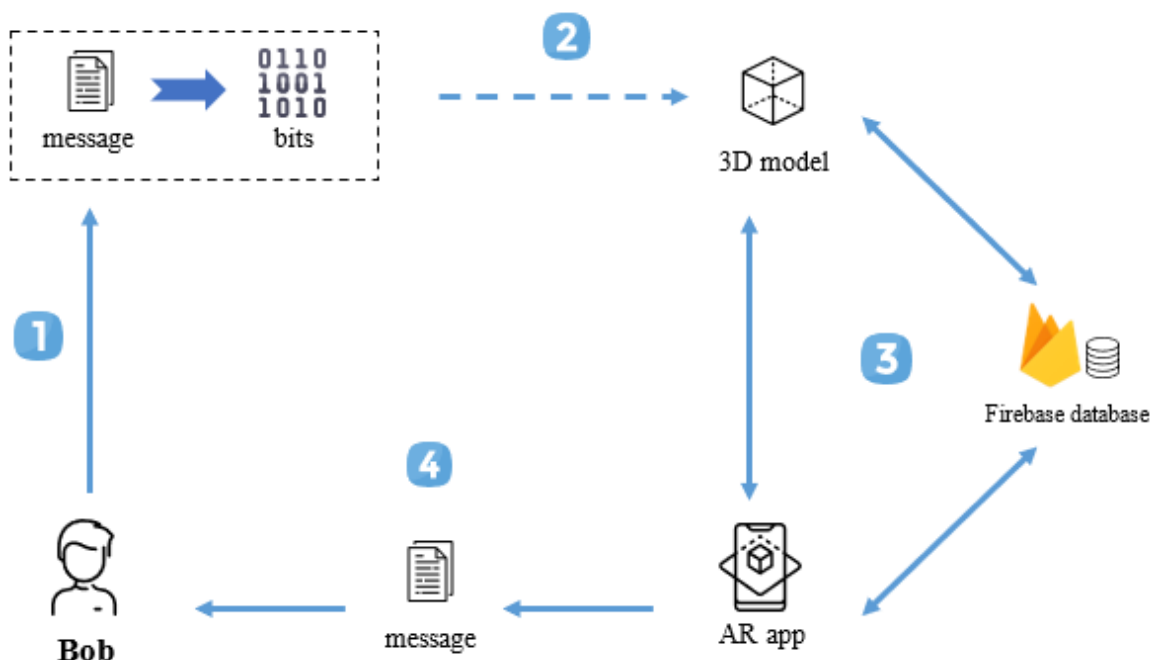
Tại điểm (2), người dùng với dữ liệu thô thập lục phân thực hiện thuật toán One Time Pad (OTP) với tệp tin nhị phân (bin file) chứa thuộc tính của đối tượng. Thông thường nếu không cấu hình hoặc đọc cấu trúc đối tượng từ tệp glTF, dữ liệu sẽ được ghi từ bộ đệm dưới cùng lên. Hạn chế sự tiếp xúc đối với các chỉ mục thuộc tính của vật thể 3D. Các chỉ mục này có vị trí tuyệt đối trong bộ đệm và quy định toàn bộ kết cấu của vật thể. Cơ bản từ vị trí đệm ban đầu đến cuối cùng của một vật

thể 3D theo trình tự: Chỉ mục, vị trí các đỉnh, hướng của các mặt đa diện (vector pháp tuyến - chiều hướng các mặt ra ngoài), độ cong mặt phẳng, màu sắc... [21].

Phân đoạn cuối cùng của tệp tin đệm (buffer) thường là màu sắc. Việc thay đổi tùy ý không theo nguyên tắc sẽ làm sai kết cấu ban đầu của vật thể.

Tại điểm (3), như trong mô hình đầy đủ (Hình 2.1) hệ thống máy chủ có hai thành phần hỗ trợ là Google Anchor Cloud và GPS API thì trong mô hình minh họa đơn thể chỉ giữ lại máy chủ cơ sở dữ liệu Firebase. Đối tượng 3D sau khi được dung nạp thông điệp sẽ lưu trữ trên cơ sở dữ liệu đám mây – lắng nghe sự kiện trong thời gian thực. Thông qua bộ xử lý sự kiện trong ứng dụng, tệp nhị phân được tải về từ máy chủ và hiển thị đối tượng 3D ra bề mặt thật. Ba thành phần: Vật thể 3D, máy chủ và hệ thống xử lý sự kiện tại local là ba thành tố tương tác qua lại với nhau trong quá trình ứng dụng thực thi.

Tại điểm cuối (4), đối tượng chứa tệp tin nhị phân mới được tải về từ phía máy chủ. Đối tượng được hiển thị ra màn hình, đồng thời dữ liệu bộ đệm được lưu lại. Thực hiện XOR bit với tệp tin gốc kết hợp đảo ngược thông điệp ta thu được thông điệp ban đầu của người dùng. Quá trình quay lại điểm đầu (1), khi người dùng có một thông điệp mới được nhập vào thì quá trình diễn ra tương tự như các bước vừa trình bày.



Hình 3.3. Mô hình triển khai giấu tin cơ bản đơn đối tượng

3.3. Xây dựng ứng dụng minh họa

3.3.1. Giới thiệu ứng dụng

Trên cơ sở thành lập nghiên cứu các chức năng cơ bản ban đầu, làm nền tảng cho việc phát triển thêm các chức năng mã hóa và chia sẻ tài nguyên; ứng dụng nền tảng Android được tạo ra với chức năng và thao tác cơ bản: “Sử dụng định dạng glTF trong ARCore/Sceneform cho phép người dùng nhập thông điệp và ẩn nó vào tệp tin nhị phân. Có khả năng kết xuất trong thời gian thực thi ứng dụng” [22].

Trong đó, ứng dụng có thể đáp ứng một số chức năng, cơ bản:

- Đăng nhập xác thực tài khoản bằng Firebase. Sử dụng Google account.
- Nhập tin nhắn, tiền xử lý trước khi lưu trữ đám mây.
- Nhúng thông điệp vào đối tượng 3D.
- Lưu trữ tệp tin nhị phân chứa đối tượng 3D của người dùng.
- Tải đối tượng trong thời gian thực và kết xuất tại điểm người dùng chọn trong thời gian ứng dụng chạy (runtime).
- Hiện thị đối tượng và hiện thị tin nhắn đã tạo trước đó.

Về chức năng mở rộng, có những đáp ứng sau:

- Phân tích cấu trúc đối tượng (tệp tin JSON glTF).
- Kết hợp khóa bí mật và cặp khóa bất đối xứng trong xử lý nhúng tin.
- Mở rộng các chế độ nhúng tin.
- Cho phép tải lên các tệp tin đối tượng có textures (chi tiết).
- Lưu trữ khóa và hàm trên máy chủ để thực hiện mã hóa an toàn.
- Kết hợp Google Anchor Cloud, cho phép chia sẻ vị trí điểm đặt vật thể trong giới hạn không gian thực.
- Kết hợp GPS API cho phép chia sẻ vị trí người dùng để giới hạn không gian quét vật thể kèm theo khả năng mở rộng không gian bối cảnh.
- Giao diện thân thiện cho phép các tài khoản xác thực lẫn nhau và trao đổi thông tin cho nhau.
- Xử lý bất đồng bộ trong việc truyền và tải thông tin từ máy chủ. Đem đến cho người dùng những trải nghiệm tốt nhất.
- Cải thiện tốc độ xử lý và tối ưu các hàm giấu tin. Hạn chế các tình trạng giật, chậm, treo ứng dụng,...
- Cho phép người dùng tìm kiếm và áp dụng các đối tượng ba chiều có sẵn trên nguồn cung Internet hoặc tải lên từ thiết bị của mình.

3.3.2. Xây dựng hệ thống lưu trữ tập tin nhị phân mật

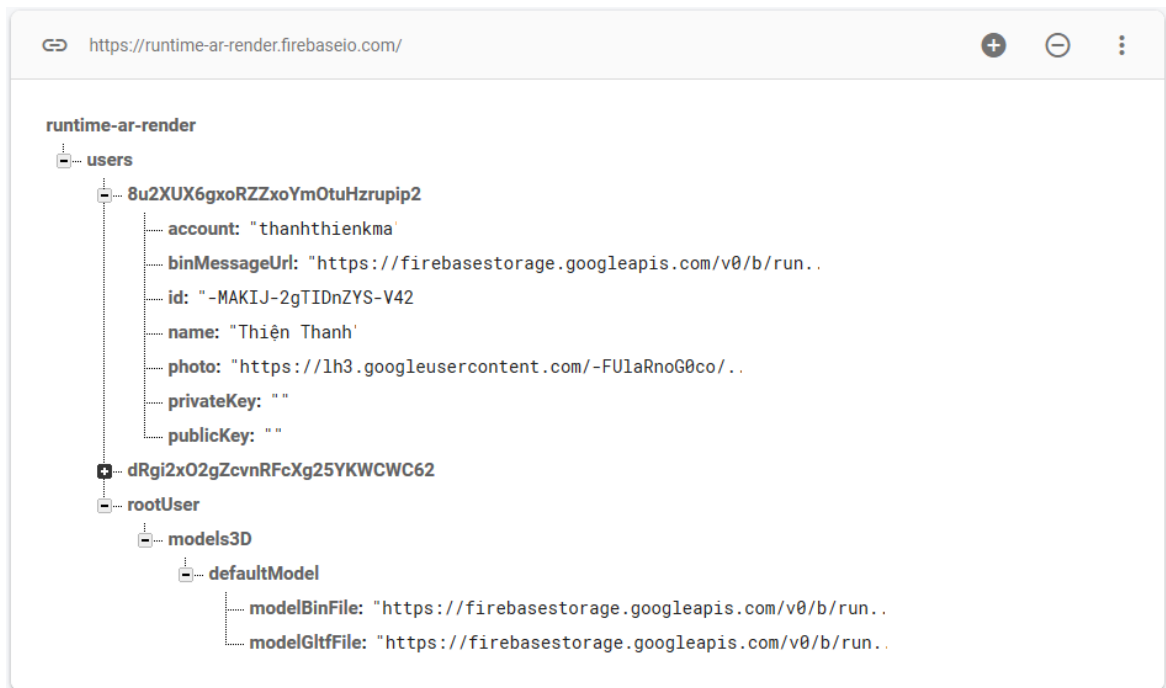
Firebase là một nền tảng máy chủ hiện đại có thể phát triển ứng dụng web hay di động, đã bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server. Giúp nhà phát triển rút ngắn thời gian và mở rộng quy mô dự án.

Trong mô phỏng này, ứng dụng sử dụng một số dịch vụ sau của Firebase như: Firebase Authentication, Firebase Storage, Firebase Realtime Database,...

Firebase Realtime Database tổ chức file dưới dạng cây thông tin. Từ một nút gốc, các nút con được tạo ra với cấp độ và mức ưu tiên thấp hơn.

Khi người dùng đăng nhập thành công vào tài khoản để sử dụng ứng dụng: “Giấu tin áp dụng AR” thì tài khoản người dùng được khởi tạo với các thông tin trống hoặc trích rút từ Google account. Mỗi đối tượng khi đăng nhập Google có chứa thông tin về UID, được sử dụng để làm khóa chính cho mỗi tài khoản đã đăng nhập.

Trong Hình 3.4, trình bày mô hình các nút trong cơ sở dữ liệu thời gian thực Firebase. Luôn luôn có một nút gốc người dùng (rootUser) lưu lại đường dẫn của đối tượng 3D mặc định. Khi cơ sở dữ liệu không tồn tại hoặc khởi tạo lại. Chương trình sẽ kiểm tra, nếu rootUser trống lập tức được khởi tạo một nút mới với đối tượng mặc định được nạp vào. Thuộc tính binMessageUrl được thay đổi mỗi khi người dùng gửi thông điệp. Đường dẫn này trỏ đến tệp tin nhị phân đã nhúng thông điệp.



Hình 3.4. Minh họa cơ sở dữ liệu thời gian thực trên database

Ngoài ra, Firebase còn cho phép lưu trữ các tệp tin trong hệ thống của mình. Bằng cách này, tệp tin mang dữ liệu người dùng được lưu lại. Thuận tiện cho quá trình chia sẻ thông điệp sau này.

Trong mô tả ở Hình 3.5, luôn tồn tại một thư mục rootUser chứa tệp tin về đối tượng 3D mặc định. Chương trình sẽ kiểm tra thư mục này, nếu không tồn tại sẽ khởi tạo và cập nhật đường dẫn trong rootUser database. Mặc khác các tệp tin người dùng cũng được lưu ở đây.

Bước đầu triển khai, chỉ cho phép người dùng lưu tệp tin nhị phân chứa thông điệp ẩn bên trong, sử dụng đối tượng gốc làm vật mang và là công cụ giải mã.

Khi người dùng nhập thông điệp mới. Thông điệp cũ được cập nhật lại bằng cách hệ thống sẽ gửi một tệp tin mới đến Firebase Storage, khi tệp tin được tải xong cơ sở dữ liệu cập nhật lại đường dẫn. Khi người dùng truy vấn thông điệp đã nhập trước đó. Tệp tin chứa thông điệp được tải lên sẽ được XOR bit cùng với tệp tin gốc của đối tượng trong rootUser, đảo ngược thông tin nhận được, giải mã thông điệp (nếu có). Song song đó, tệp tin “cube.glTF” được tải về trên bộ nhớ trong của thiết bị cùng với tệp tin “cube.bin” chứa thông điệp. Hiển thị thông báo cho người dùng khi người dùng tương tác hoặc đặt vật thể trong không gian thực.

The screenshot displays the Firebase Storage console with three views of the file hierarchy:

- View 1:** Path `gs://runtime-ar-render.appspot.com > users`. It shows a list of folders: `8u2XUX6gx0RZZxoYm0tuHzruplp2/`, `dRgi2x02gZcvmRFcXg25YKWCWC62/`, and `rootUser/`.
- View 2:** Path `gs://runtime-ar-render.appspot.com > users > rootUser > models3D > defaultModel`. It shows two files: `cube.bin` (1.01 KB) and `cube.glTF` (6.86 KB).
- View 3:** Path `gs://runtime-ar-render.appspot.com > users > 8u2XUX6gx0RZ... > models3D > defaultModel`. It shows one file: `cube.bin` (1.01 KB).

Name	Size
8u2XUX6gx0RZZxoYm0tuHzruplp2/	—
dRgi2x02gZcvmRFcXg25YKWCWC62/	—
rootUser/	—

Name	Size
cube.bin	1.01 KB
cube.glTF	6.86 KB

Name	Size
cube.bin	1.01 KB

Hình 3.5. Firebase Storage – kho lưu trữ tệp tin

3.3.3. Thu nhận thông điệp và ẩn giấu thông tin

Ứng dụng được xây dựng trên nền tảng Android sử dụng ngôn ngữ Kotlin. Bước đầu trong định hướng dài hạn về khả năng kết nối thông điệp giữa các tài khoản với nhau là việc xây dựng hệ thống xác thực tài khoản. Sử dụng Google account để đăng nhập và quản lý thông tin [21][22].

Trước nhất, làm quen với giao diện ứng dụng; ta có hai giao diện chính/layout (trong Android gọi là activity) như sau:

- *SignInActivity*: Đây là màn hình đầu tiên khi ứng dụng lần đầu tiên khởi chạy. Khi người dùng là một tài khoản mới hoàn toàn. Chưa đăng nhập vào ứng dụng lần nào. Ứng dụng đòi hỏi quyền truy cập bằng tài khoản Google nhằm định danh người dùng, dễ dàng liên kết các dịch vụ và quản lý tài nguyên, tài khoản sau này. Màn hình tập trung chính vào chức năng đăng nhập nên duy chỉ có một nút đăng nhập trên màn hình. Đây là widget được cung cấp bởi thư viện Google.
- *MainActivity*: Màn hình tương tác chính của ứng dụng. Tại đây sau khi người dùng đăng nhập tài khoản Google tại *SignInActivity*, *MainActivity* sẽ kiểm tra xem tài khoản đăng nhập có hợp lệ hoặc đã đăng nhập chưa rồi mới chuyển sang thực hiện các chức năng. Tại đây cấu trúc của layout được chia thành ba phần:
 - + *Màn hình hiển thị thực tế ảo*: *ArFragment* được cung cấp bởi thư viện của Seneform, tại đây các chức năng như: Phát hiện bề mặt đặt vật thể, gợi ý người dùng, điều hòa biên độ sáng, lưu phiên xử lý đối tượng,... được mặc định cấu hình.
 - + *Nhập dữ liệu đầu vào*: Bao gồm tương tác giữa *EditText*, cho phép người dùng nhập thông điệp của mình sau đó nhấn nút *Send* (gửi) để xử lý thông tin. Ngoài ra, chương trình có một tùy chọn là “High-bit” cho phép nhúng tin ở bit cao để cho thấy sự thay đổi cấu trúc của vật thể khi tồn tại điểm sai trong quá trình giấu tin.
 - + *Thanh hiển thị toolbar*: Hiển thị tên ứng dụng, menu có chứa một tùy chọn “Sign out” khi người dùng không muốn tiếp tục trải nghiệm với tài khoản hiện tại.



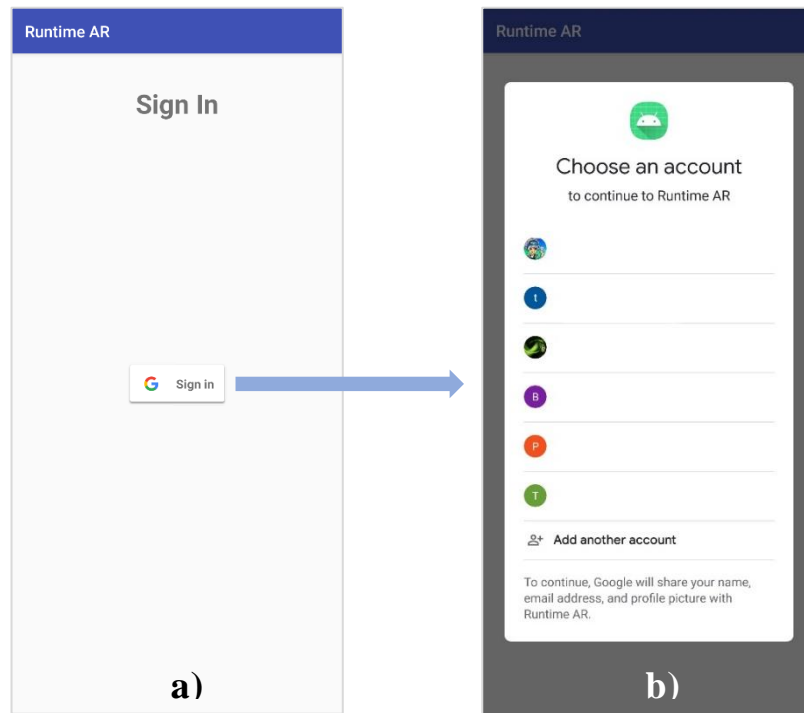
Hình 3.6. Giao diện làm việc nền tảng của ứng dụng

Phiên kết nối khi người dùng đăng nhập vào Google được duy trì khá lâu mà không bắt thực hiện đăng nhập trong thời gian dài. Khi kết nối được duy trì, ứng dụng sẽ ngay lập tức chuyển màn hình đến MainActivity khi ứng dụng được yêu cầu truy cập.

Để cụ thể hóa các thành phần chức năng, cũng như có được cái nhìn tổng quan về quá trình hoạt động của ứng dụng. Khởi chạy chương trình và phân tích các phương thức/chức năng quan trọng cần được nhắc đến.

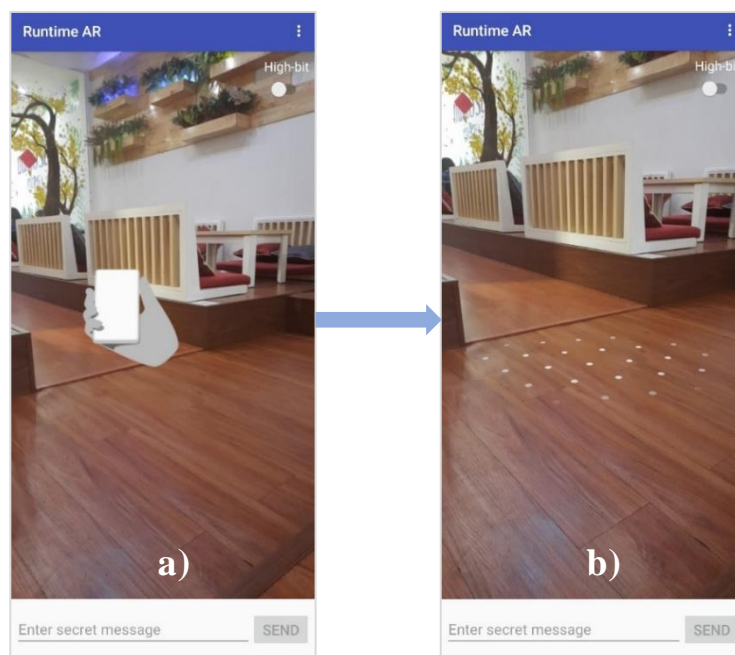
Khởi chạy ứng dụng:

Trong Hình 3.7.a, tại màn hình khởi động của ứng dụng; người dùng chạm vào nút “Sign In” trên màn hình. Màn hình sẽ hiển thị trên nền một cửa sổ đăng nhập được cung cấp bởi Google API (Hình 3.7.b). Hiển thị một số các tài khoản gần đây người dùng sử dụng trên thiết bị của mình. Người dùng có thể tùy chọn tài khoản sử dụng hoặc thêm mới một tài khoản khác.



Hình 3.7. Giao diện đăng nhập tài khoản

Sau khi người dùng đăng nhập thành công với tài khoản Google của mình. MainActivity hiện ra, tại đây là phần màn hình chính tương tác AR. Hình 3.8.a, một hình ảnh chuyển động gợi ý người dùng di chuyển điện thoại di động một vòng tròn theo hướng vuông góc với mặt phẳng nhằm giúp các thành phần cảm biến phát hiện được bề mặt phẳng có thể đặt vật thể (Lưu ý: tránh những bề mặt gồ ghề, dốc và các bề mặt trong suốt hay có màu trắng,...) [20][22].



Hình 3.8. Quét và phát hiện bề mặt đặt vật thể 3D

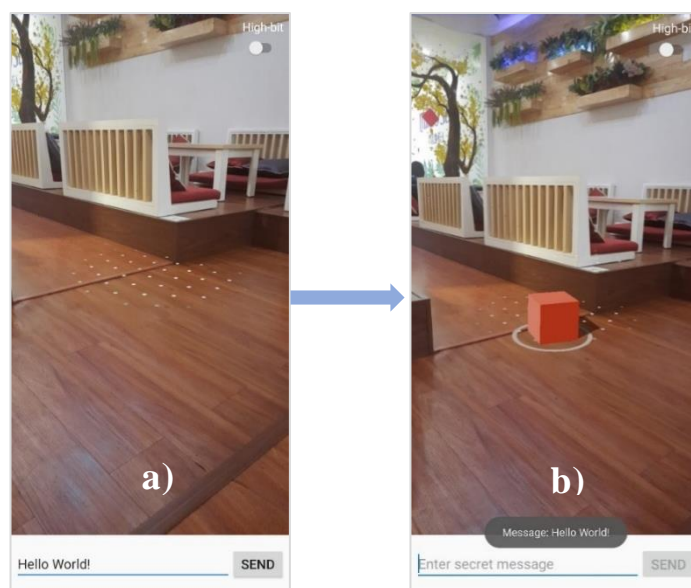
Khi phát hiện bề mặt thành công, chuyển động bàn tay biến mất, trên bề mặt phẳng được phát hiện xuất hiện bề mặt với những điểm chấm màu trắng; dấu hiệu thông báo bề mặt được phát hiện (Hình 3.8.b). Bằng việc chạm vào bề mặt, ta hiển thị được hình ảnh 3D trong thế giới thực (ứng dụng sử dụng hình ảnh mặc định được định nghĩa trước).



Hình 3.9. Đặt vật thể 3D trong thế giới thật

Trong Hình 3.9, đối tượng được đặt trên bề mặt có tương quan về độ sáng và bóng đổ. Nhờ vào sự hỗ trợ của Sceneform, một API bậc cao trên nền tảng ARCore, ta không cần quan tâm nhiều đến khả năng tương tác và học hỏi của AR với các thành phần bên ngoài của thế giới thật. Các thao tác trên là những thao tác cơ bản của một ứng dụng AR.

Để tiện cho việc phân tích và thấy được mức độ ảnh hưởng đến cấu trúc của đối tượng 3D. Ta chọn một mô hình đơn giản hơn – phục vụ cho mục đích nghiên cứu/minh họa.



Hình 3.10. Giấu tin vào đối tượng 3D và hiển thị trên bề mặt thật

Trước hết, ta tìm hiểu hiệu quả giấu tin của ứng dụng trước khi đi vào phân tích nó. Nhập một đoạn thông điệp tại khung EditText và ấn “Send” để gửi thông điệp xử lý (Hình 3.10.a). Sau đó, ta đặt lại một vật thể mới trên bề mặt đã phát hiện trước đó (Lưu ý: Không bật chế độ High-bit). Thao tác xong, ta nhận được một vật thể 3D không có đặc điểm hay thay đổi gì khác so với đối tượng ban đầu (Hình 3.10.b). Tuy nhiên, đối tượng được ghi nhận là đã có nhúng tin và hiển thị tin nhắn vừa giải mã từ vật thể được đặt ra trên màn hình.

Vậy chuyện gì đã xảy ra đối với thông tin và vật thể đã nhập vào. Để tìm hiểu điều này, ta phân tích tệp tin vật thể 3D đã kết xuất trong ví dụ trên: “cube.glTF” và “cube.bin”

Trong ví dụ trên, đối tượng đơn giản chỉ có một tệp tin JSON (*.glTF) lưu thông tin cấu trúc vật thể và tệp tin nhị phân lưu bộ đệm tương ứng (Hình 3.11).



Hình 3.11. Tài nguyên của mô hình 3D mô phỏng

Ta xem xét tệp tài nguyên “cube.glTF”, lược bỏ một vài yếu tố không then chốt. Ta lọc được một số thông tin về liên hệ cấu trúc của đối tượng trên:

```

"accessors" : [
  {
    "bufferView" : 2,
    "byteOffset" : 0,
    "componentType" : 5126,
    "count" : 24,
    "max" : [ 1, 0.09387587010860443, 0.01188133470714092, 1 ],
    "min" : [ 1, 0.09387587010860443, 0.01188133470714092, 1 ],
    "name" : "accessor_color_m0-PolyPaper14_-26232",
    "type" : "VEC4"
  },
  {
    "bufferView" : 0,
    "byteOffset" : 0,
    "componentType" : 5123,
    "count" : 36,
    "max" : [ 23 ],
    "min" : [ 0 ],
    "name" : "accessor_indices_0_m0-PolyPaper14_-26232",
    "type" : "SCALAR"
  },
  {
    "bufferView" : 1,
    "byteOffset" : 288,
    "componentType" : 5126,
    "count" : 24,
    "max" : [ 1, 1, 1 ],
    "min" : [ -1, -1, -1 ],
    "name" : "accessor_normal_m0-PolyPaper14_-26232",
    "type" : "VEC3"
  },
  {
    "bufferView" : 1,
    "byteOffset" : 0,
    "componentType" : 5126,
    "count" : 24,
    "max" : [ -0.7000008, 1.309997, 0.5100036999999999 ],
    "min" : [ -0.7800009, 1.229998, 0.4300037 ],
    "name" : "accessor_position_m0-PolyPaper14_-26232",
    "type" : "VEC3"
  }
],

"bufferViews" : [
  {
    "buffer" : 0,
    "byteLength" : 72,
    "byteOffset" : 0,
    "name" : "ushortBufferView",
    "target" : 34963
  },
  {
    "buffer" : 0,
    "byteLength" : 576,
    "byteOffset" : 72,
    "byteStride" : 12,
    "name" : "vec3BufferView",
    "target" : 34962
  },
  {
    "buffer" : 0,
    "byteLength" : 384,
    "byteOffset" : 648,
    "byteStride" : 16,
    "name" : "vec4BufferView",
    "target" : 34962
  }
],
"buffers" : [
  {
    "byteLength" : 1032,
    "name" : "cube",
    "uri" : "cube.bin"
  }
]

```

Hình 3.12. Các thành phần cơ bản trong cấu trúc JSON – glTF

Như đã trình bày ở phần 2.3.2, có ba thành phần cấu trúc không thể thiếu đối với đối tượng 3D là: accessors, buffers và bufferViews trong file JSON – glTF. Các đối thành phần này ánh xạ đến bộ nhớ đệm trong tệp tin nhị phân: “cube.bin”.

- *buffers*:

- + *byteLength*: Dung lượng của tệp tin nhị phân (1032).
- + *name*: Tên của tệp tin chứa bộ đệm (cube).
- + *uri*: Đường dẫn đến tệp tin nhị phân (cube.bin – cùng nhánh thư mục).

Buffers cho biết thông tin cơ bản của tệp tin nhị phân (*.bin) và gọi đến nó khi các thành phần cần trích xuất thông tin từ bộ đệm. Có thể có nhiều buffer một tệp tin glTF. BufferViews có nhiều phân lớp con, trong ví dụ trên BufferViews

được chia theo kiểu dữ liệu lưu trữ, có ba kiểu dữ liệu lưu trữ hợp lệ là: SCALARS, VECTORS và MATRICES.

- *bufferViews*:
 - + *buffer*: Cho thấy bufferView thuộc buffer nào (0 – thứ tự luôn từ 0).
 - + *bufferLength*: Cho biết độ dài (tính bằng byte) của mỗi accessor tương ứng khi ánh xạ vào.
 - + *byteOffset*: Chỉ ra vị trí bắt đầu của bufferView trong buffers và kết thúc bằng cách cộng thêm bufferLength vào byteOffset.
 - + *name*: Tên bufferView. Ở đây, tên được phân cấu theo kiểu dữ liệu: ushortBufferView (SCALARS), vec3BufferView (VECTORS) và vec4BufferView(MATRICES).
- *accessors*:
 - + *bufferView*: Chỉ số với bufferView tương ứng.
 - + *byteOffset*: Chỉ ra vị trí bắt đầu của accessor trong bufferView.
 - + *componentType*: Chỉ ra kiểu giá trị đối với kiểu dữ liệu type. Có hai kiểu giá trị là 5126 (FLOAT) và 5123 (UNSIGNED_SHORT).
 - + *max/min*: Định mức giới hạn của giá trị. Thuộc tính này thay đổi theo componentType tương ứng.
 - + *type*: Kiểu dữ liệu. Có ba kiểu giá trị đã nêu.
 - + *name*: Tên của các accessor, trong đối tượng mẫu này, tác giả đã đặt tên các thuộc tính tương ứng cho các phân đoạn bộ đệm khác nhau như: *accessor_color* (phân đệm quy định màu), *accessor_indices* (phân đệm chỉ số - dữ liệu còn lại phải tuân thủ theo các sắp xếp này), *accessor_normal* (phân đệm quy định hướng của bề mặt của vật thể), *accessor_position* (phân đệm vị trí của vật thể so với tọa độ gốc trong không gian thực).

Dựa trên phân tích cấu trúc trên. Áp dụng vào các phân đệm nhị phân ta có bảng sơ khảo kết quả của các accessor so với toàn bộ tệp tin nhị phân như sau:

Accessor	BufferView	ByteOffset	Buffer fragment
Indices	0	0	0300 0100 0000 0300 0200 0100 0700 0500 0400 0700 0600 0500 0b00 0900 0800 0b00 0a00 0900 0f00 0d00 0c00 0f00 0e00 0d00 1300 1100 1000 1300 1200 1100 1700 1500 1400 1700 1600 1500
Position	1	72	23ae 47bf 8f70 9d3f 7329 dc3e 23ae 47bf ffad a73f 7429 dc3e 23ae 47bf ffad a73f 9b8f 023f 23ae 47bf 8f70 9d3f 9b8f 023f 4133 33bf 8f70 9d3f 7329 dc3e 4133 33bf 8f70 9d3f 9b8f 023f ...
Normal	1	360	ffff 7fbf 0000 0000 0000 0080 0000 80bf 0000 0000 0000 0080 0000 80bf 0000 0000 0000 0080 0000 80bf 0000 0000 0000 0080 ffff 7f3f 0000 0000 0000 0080 0000 803f 0000 0000 0000 0080 ...
Color	2	648	0000 803f 0000 0080 0000 803f fe41 c03d eea9 423c 0000 803f 0000 803f fe41 c03d eea9 423c 0000 803f 0000 803f fe41 c03d eea9 423c 0000 803f 0000 803f fe41 c03d eea9 423c 0000 803f 0000 803f fe41 c03d ...
			fea9 423c 0000 803f 1000 803f fe41 c03d fea9 423c 0000 803f 1000 803f fe41 c03d eea9 423c 1000 803f 1000 803f fe41 c03d eea9 423c 0000 803f 1000 803f fe41 c03d eea9 423c 1000 803f

Hình 3.13. Bảng khái quát các phân đoạn đệm với accessor tương ứng

Thông tin thêm:

- Kiểu giá trị trong tệp nhị phân được biểu diễn dưới dạng chuẩn hóa IEEE 754 (thuận tiện cho việc lưu trữ số thực dấu chấm động).
- Giá trị được lưu ở kiến trúc 64 bits, gồm một cụm 8 bytes “xxxx xxxx” lưu kết quả của một giá trị.
- Mỗi indices đại diện cho chỉ số của các đỉnh có trong mô hình, mỗi chỉ số chỉ đại diện bởi một cụm 4 bytes “xxxx”. Ở mô hình lập phương có 36 đỉnh trùng lặp ở mỗi mặt (tính cả đỉnh trung tâm hình khối) nhưng chỉ kết xuất thành 24 mặt tam giác.
- Dữ liệu trong tệp nhị phân (*.bin) theo Little Endian, nghĩa là các bit trọng yếu sẽ nằm về bên phải cùng của tổ chức dữ liệu (MSB – Most Significant Bit). Các bit thứ yếu sẽ nằm về bên trái cùng của tổ chức dữ liệu (LSB –

Least Significant Bit). Năm được yếu tố này giúp truy cập được hướng tổ chức nội dung của tệp tin nhị phân (*.bin) nhằm nhúng thông tin chính xác vào các bit thứ yếu, để đối tượng không bị thay đổi kết cấu nguyên thủy của nó.

Có bốn phương pháp nhúng thông tin:

- *Phương pháp 1*: Nhúng thông tin từ vị trí cuối cùng của tệp tin.
- *Phương pháp 2*: Nhúng thông tin từ vị trí cuối cùng của accessor xác định.
- *Phương pháp 3*: Nhúng thông tin xen kẽ giữa các accessor.
- *Phương pháp 4*: Nhúng thông tin ngẫu nhiên giữa các accessor.

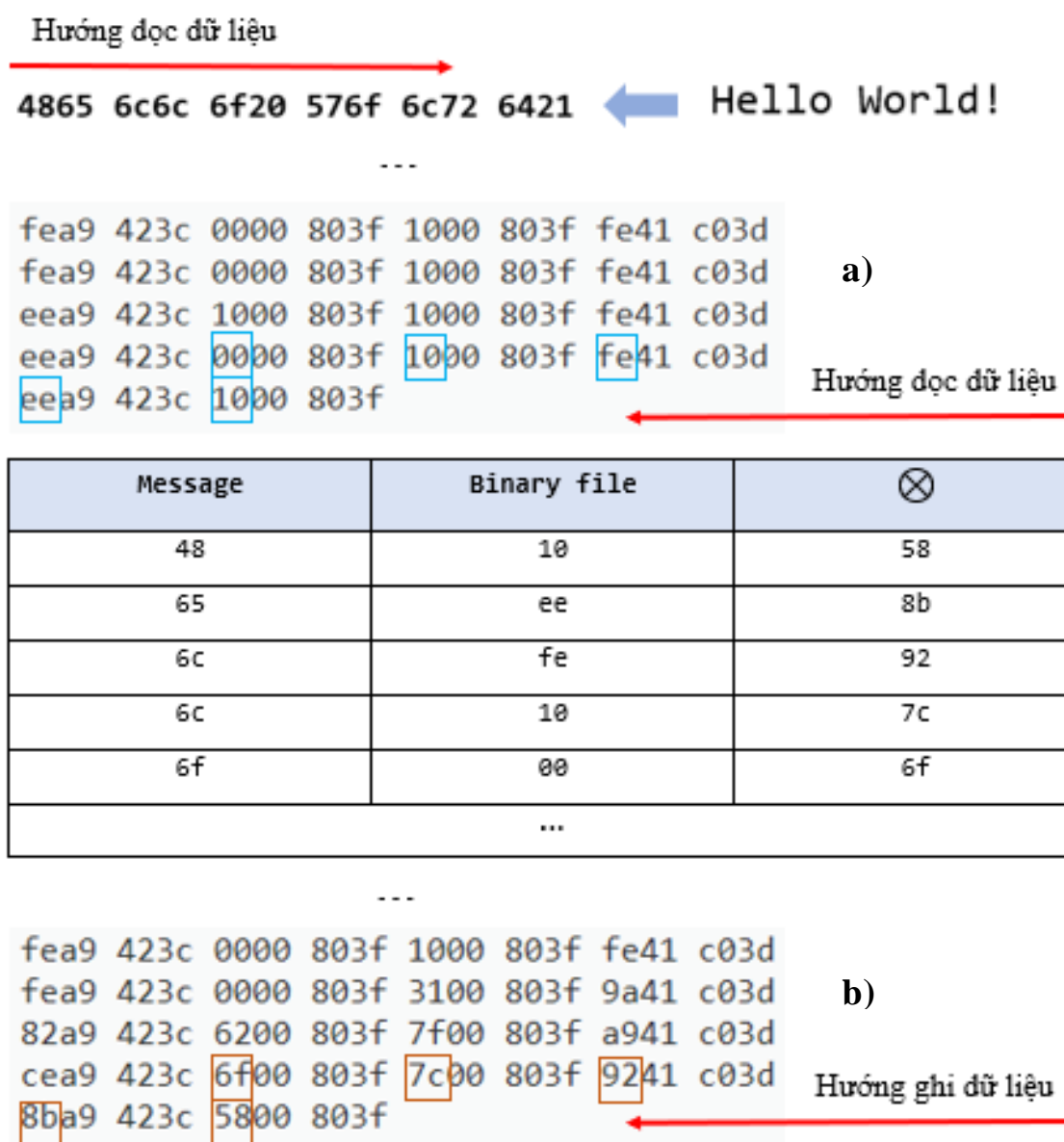
Trong giới hạn báo cáo, chỉ trình bày tập trung vào phương pháp thứ nhất.

Theo như mô phỏng của Hình 2.8, ví dụ mô phỏng tại thiết bị thật (Hình 3.8) kèm theo cấu trúc tệp tin (Hình 3.12), ta có nguyên lý giấu tin tại Hình 3.8 như sau:

Sử dụng phương pháp thứ nhất – nhúng thông tin từ vị trí cuối cùng của tệp tin. Do kiến trúc dữ liệu tuân theo Little Endian nên tính từ bytes cuối cùng đến bytes thứ yếu LSB là 4 bytes. Vậy cứ cách một khoảng là 4 bytes thì khi ghi nội dung, cấu trúc thuộc tính của đối tượng không bị thay đổi. Nếu khoảng này bị lệch hoặc vị trí ban đầu không nằm ở vị trí 4 thứ 4 thì kết cấu đối tượng rất dễ chịu ảnh hưởng.

Đối với đối tượng minh họa là “cube.bin” thì accessor cuối cùng của tệp tin đệm là màu sắc. Khả năng lệch bytes khiến màu sắc chịu ảnh hưởng nhất so với các thuộc tính khác.

Trong Hình 3.14, thông điệp người dùng nhập vào trong ứng dụng Hình 2.8 là “Hello World!” chuyển thành thập lục phân là: 4865 – 6c6c – 6f20 – 576f – 6c72 – 6421. Nếu trích các phần bytes thứ yếu từ tệp tin đệm ta có chuỗi thập lục phân tương ứng với chuỗi thông điệp trên: 10ee – fe10 – 00ee – fe10 – 10ee – fe10 (Hình 3.14.a). Thực hiện phép XOR bit, ta có chuỗi bytes tương ứng: 588b – 927c – 6fce – a97f – 6282 – 9a31 (Hình 3.14.b).



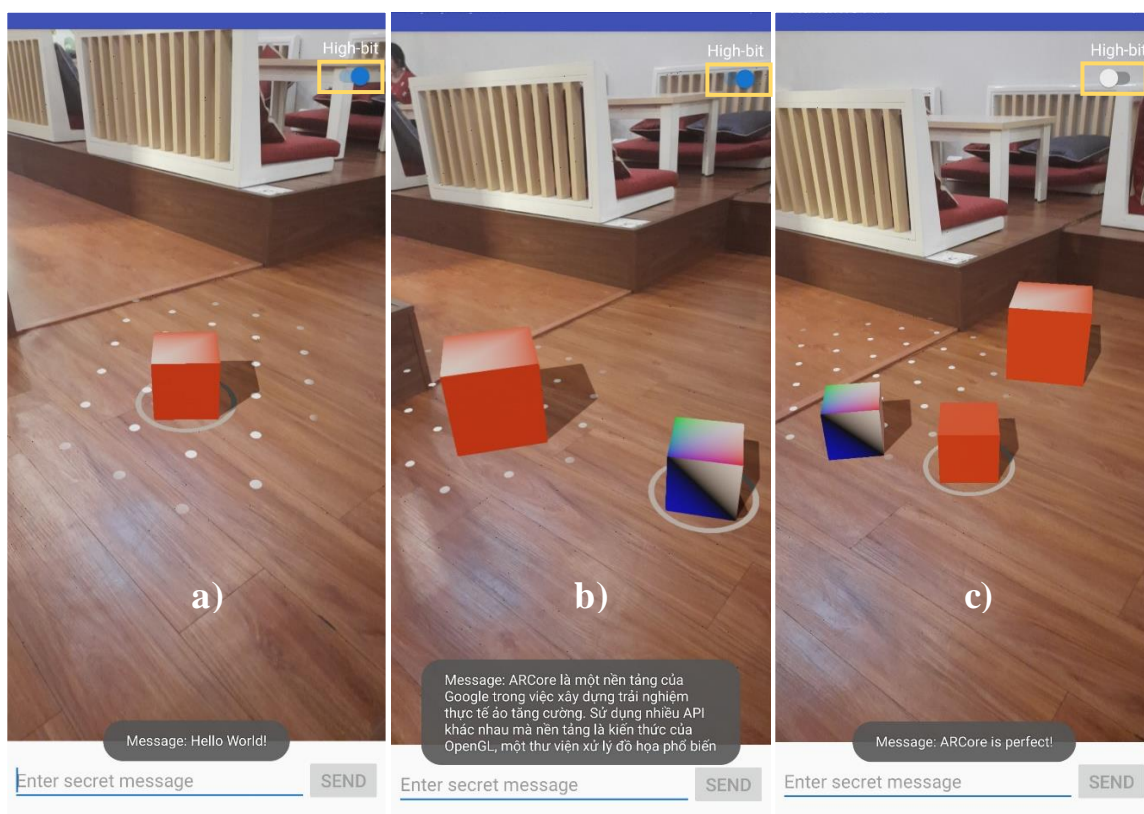
Hình 3.14. Nguyên tắc đọc ghi dữ liệu nhị phân trong bộ đệm đối tượng 3D

Minh họa khả năng tác động của việc giấu tin bit cao với giấu tin bit thấp. Ta thử nghiệm với ứng dụng để so sánh kết quả thay đổi như thế nào.

Trong Hình 3.15.a, minh họa tác động khi ứng dụng bật chế độ mã hóa ở bit cao. Ta thấy so với mô hình 3D ban đầu, màu sắc của khối lập phương bị thay đổi. Với thông tin nhập vào ngắn thì khoảng màu thay đổi không quá nhiều (chỉ xuất hiện một vệt trắng).

Với Hình 3.15.b, khi nhập một thông điệp dài hơn; màu sắc của khối lập phương thay đổi nhiều hơn. Màu sắc xuất phát từ các đỉnh của mô hình như một đơn vị ảnh hưởng nhỏ hơn.

Với Hình 3.15.c, khi tắt chế độ ẩn tin bit cao. Ta thấy rằng thông điệp không làm thay đổi kết cấu hay màu sắc khối lập phương.



Hình 3.15. So sánh giữa giấu tin bit cao với giấu tin bit thấp

Từ việc chứng minh được khả năng nhúng tin dựa trên nền tảng ARCore kết hợp với định dạng glTF là hoàn toàn khả thi và mang lại kết quả như mong đợi. Để mở rộng quy mô hệ thống và lưu trữ mô hình ba chiều có chứa thông tin mật của người dùng, ta kết hợp thông tin vật thể hệ thống máy chủ đám mây Firebase đã trình bày trong phần 3.3.2.

Ta có thể thấy kết quả thực nghiệm được hiển thị ở Hình 3.15. Đó chưa phải là kết quả cuối cùng của ứng dụng. Trong phạm vi mô phỏng báo cáo, kết quả trình bày khả năng các vật thể giấu tin có thể được đặt cùng nhau trong một môi trường. Thông điệp được ẩn giấu và mã hóa phụ thuộc vào việc nhúng các bit sao cho khoảng cách các bit nhúng là không đổi. Vị trí nhúng đảm bảo rằng bit được nhúng là bit có trọng yếu thấp (LSB) [1][17].

3.3.4. Phân tích khả năng an toàn của thông điệp và phương hướng phát triển

Mô hình ứng dụng “Giấu tin áp dụng thực tế ảo AR” trở nên an toàn hơn trong quá trình chia sẻ thông tin khi kết hợp với hệ thống máy chủ Firebase. Lưu trữ và thực hiện tính toán với khóa đảm bảo rằng khả năng thông tin bị bại lộ thấp.

- *Phân tích an toàn:*
 - Thông điệp được ẩn giấu bằng nhiều chế độ khác nhau. Khả năng xáo trộn thông tin cao.

- Kẻ tấn công nếu không lấy được thông tin đối tượng gốc, trong mô phỏng vừa rồi sử dụng “cube.glTF” và “cube.bin”. Nếu kẻ tấn công không sở hữu hai tập tin này thì khả năng chiếm được thông điệp khá thấp do thực hiện thuật toán OTP để lấy thông tin mà tệp tin nhị phân là chìa khóa.
- Kết hợp khóa phiên, đảm bảo trong mỗi phiên kết nối có một khóa bí mật khác nhau. Mã hóa lần nữa với khóa công khai phía người nhận.
- Kẻ tấn công không có khóa riêng của người nhận thì không thể giải mã được khóa phiên. Ta có thể xây dựng được nhiều khả năng ẩn giấu khóa bí mật kèm theo, tăng độ khó của bài toán giấu tin.
- Mặc khác, việc không thay đổi kích thước tệp tin nhị phân sau nhúng tin. Nếu bằng mắt thường cũng không nhìn ra bất kỳ thay đổi nào của đối tượng 3D cả. Việc kẻ tấn công không sở hữu được vị trí đặt vật thể là một bất lợi trong quá trình tiếp cận thông điệp.
- *Phương hướng phát triển:*
 - Nghiên cứu phát triển thêm các định dạng khác làm vật mang thông điệp trong ứng dụng AR.
 - Xây dựng hệ thống chia sẻ đối tượng 3D trong không gian thực, sử dụng Google Anchor và GPS API.
 - Nghiên cứu và phát triển hệ thống phát hiện giấu tin sử dụng AR.
 - Thiết lập hệ thống tường lửa để đảm bảo an toàn đầu – cuối cho dữ liệu của người dùng.
 - Nghiên cứu khả năng trích xuất tập tin từ vật thể đã được đặt trong môi trường thực tế ảo.

3.4. Kết luận chương 3

Chương trình minh họa ứng dụng: “Kỹ thuật giấu tin áp dụng thực tế ảo tăng cường AR” đã xây dựng tương đối thành công. Về mặt thực nghiệm, đã phân tích và mô phỏng được quy trình giấu tin và các bước xây dựng mô hình máy chủ an toàn. Chương trình cho thấy được người thực hiện báo cáo đã nắm được các quy tắc trong an toàn thông tin nói chung và quy trình ẩn mã nói riêng. Ngoài ra, giúp mọi người hiểu thêm về một phương pháp mới trong bài toán kỹ thuật giấu tin hiện đại, giúp họ có hướng tiếp cận nhanh chóng và làm chủ ứng dụng của mình.

KẾT LUẬN

Ba chương của đề án đã thể hiện được rằng những mục tiêu đặt ra khi thực hiện đề án đều đã đạt được. Cụ thể:

Chương 1 đã khái quát được quá trình hình thành và lịch sử phát triển của công nghệ thực tế ảo. Cho thấy được tiềm năng chiếm lĩnh thị trường của nó. Bằng việc vận dụng các kiến thức đã học cũng như trải qua quá trình tìm tòi và đúc kết, chương này đã trình bày được những kỹ thuật chủ yếu trong giấu tin như: *Kỹ thuật giấu tin LSB, kỹ thuật giấu tin Wu-Lee,...* Trình bày được mô hình giấu tin lưới ba chiều và giấu tin trong video – có tiềm năng ứng dụng trong tương lai vào công nghệ thực tế ảo. Nêu lên được các vấn đề an toàn trong sử dụng ứng dụng thực tế ảo và tiềm năng giấu tin trong các đối tượng ba chiều.

Chương 2 thể hiện được yêu cầu cần thiết và những thử thách đối với việc xây dựng ứng dụng thực tế ảo. Từ đó, xây dựng hệ thống tổng quan mô hình triển khai giấu tin trong ứng dụng đó. Đã chỉ ra được khả năng tích hợp các thuật toán mật mã để mở rộng chức năng bảo mật của chương trình. Giới thiệu nền tảng sử dụng là ARCore/Sceneform. Vật mang tin quan trọng trong đề án này là định dạng glTF – làm rõ cấu trúc cơ bản của định dạng, phục vụ quá trình nhúng tin và trích xuất thông tin.

Chương 3 đã trình bày được ứng dụng mô phỏng giấu tin đơn người dùng. Vận dụng kiến thức đã nêu ở chương 1 và chương 2. Chương này đã phân tích luồng ứng dụng để chỉ ra quá trình nhúng tin và giải mã như thế nào. Người dùng có thể thấy được hiệu quả giữa nhúng tin bit cao và nhúng tin bit thấp.

Trong quá trình triển khai thực nghiệm, đề án đã xây dựng được mô hình mô phỏng giấu tin trên nền tảng thực tế ảo. Tuy nhiên trên thực tế, có nhiều định dạng vật thể 3D được hỗ trợ nhưng không phải định dạng nào cũng có thể xử lý dữ liệu trong thời gian ứng dụng thực thi. Kèm theo đó khả năng trích rút ngược tệp tin của đối tượng 3D khi đặt vật thể là bất khả thi nên thám mã ngoài thực tế là một điều khá khó thực hiện. Việc giải quyết nhưng điểm tồn tại cũng chính là hướng phát triển đề án trong tương lai.

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Ngọc Cương, *Kỹ thuật giấu tin*, Học viện Kỹ thuật Mật mã, 2016.
- [2]. Jennifer Whyte and Dragana Nikolic, *Virtual Reality and the Built Enviroment*, Routledge, 2018.
- [3]. Stuart Wilson, *Unreal Steganography: Using a VR application as a steganography carrier*, Đại học An ninh máy tính và Điều tra số Sheffield Hallam, 2019.
- [4]. Dieter Schmalstieg và Tobias Hollerer, *Augmented Reality Principles and Practice*, Nhà xuất bản giáo dục Pearson, 2016.
- [5]. Tomasz Mazuryk and Michael Gervautz, *Virtual Reality History, Applications, Technology and Future*, Học viện Đồ họa máy tính Úc, 12/1999.
- [6]. Lucio Tommaso De Paolis, *Virtual and Augmented Reality Applications*, Viện nghiên cứu thực tế ảo và thực tế ảo tăng cường – Đại học Salento Ý, 03/2015.
- [7]. ThS Trần Quốc Trung, *Ứng dụng công nghệ “thực tế ảo” trong lĩnh vực giáo dục, đào tạo*, Tạp chí Lý luận chính trị số 10 – 2016.
- [8]. Nick Dauchot, *Virtual Reality: The Good, Bad and Ugly*, Medium, 10/2018.
- [9]. Rubin Peter, *Future Presence: How Virtual Reality Is Changing Human Connection, Intimacy, and the Limits of Ordinary Life*, HarperOne, 2018.
- [10]. Mikko O.Heikkinen, Kai Havukainen, Jukka Holm, Antti Eronen, Timo Kosonen, *Method and device for augmented reality message hiding and revealing*, Bằng sáng chế Hoa Kỳ, 2007.
- [11]. ChuanlongLi, Xingming Sun và Yuqian Li, *Information hiding based on Augmented Reality*, Viện Sinh toán học và Kỹ thuật, 05/2019.
- [12]. Thái Duy Quý, *Một đề xuất sử dụng lưới 3D khép kín để giấu tin*, Tạp chí khoa học đại học Đà Lạt Tập 6 Số 2, 16/03/2016.
- [13]. Chao-Hung Lin, Min-Wen Chao, Jyun-Yuan Chen, Cheng-Wei Yu và Wei-Yen Hsu, *A high-capacity distortion-free information hiding alorthm for 3D polygon models*, Tạp chí quốc tế về điện toán sáng tạo, thông tin và kiểm soát Tập 9 Số 3, 03/2013.
- [14]. Walter Bender, *Techniques for Data Hiding*, IBM Systems Journal, 2014.

- [15]. Nguyễn Văn Tảo, *Một số thuật toán giấu tin và áp dụng giấu tin mật trong ảnh*, Tạp chí Khoa học và Công nghệ, số 4(44), Tập 2.
- [16]. Nguyễn Xuân Huy, Trần Quốc Dũng, *Một thuật toán thủy vân ảnh trên miền DCT - An Image Watermarking Algorithm Using DCT Domain*, Kỷ yếu Hội thảo Quốc gia: Một số vấn đề chọn lọc của Công nghệ Thông tin, Thái Nguyên, 29-31/08/2003, NXB Khoa học Kỹ thuật Hà Nội, 2005.
- [17]. Nguyễn Diễm Hương, *Kỹ thuật giấu tin trên K bit LSB của ảnh*, Đồ án tốt nghiệp trường Đại học Dân Lập Hải Phòng, 2012.
- [18]. Phạm Trung Thành, *Tìm hiểu các kỹ thuật giấu tin trong ảnh*, Luận văn thạc sĩ chuyên ngành khoa học máy tính Đại học Thái Nguyên, 2011.
- [19]. GS. Nguyễn Bình, ThS. Hoàng Thu Phương, *Giáo trình Cơ sở lý thuyết Mật mã*, Học viện Kỹ thuật Mật mã, 2013.
- [20]. Google Developer Team, *ARCore Documentations*, Google Developer Team, 2019, Online: <https://developers.google.com/ar/develop/java/quickstart>.
- [21]. The Khronous Group Inc, *glTF Documentations*, The Khronous Group Inc, 2017, Online: <https://www.khronos.org/glTF/>.
- [22]. Jens Grubert, Dr. Raphael Grasset, *Augmented Reality for Android Application Development*, Packt Publishing, 11/2013.

PHỤ LỤC

Mã nguồn mô phỏng chương trình giấu tin trong thực tế ảo tăng cường AR trong *Chương 3 (bản rút gọn)*:

1. Các hàm xử lý trong chương trình

Các hàm chuyển đổi định dạng dữ liệu, xử lý định dạng glTF và thuật toán giấu tin trong tệp GLTFHandler.kt:

```
class GLTFHandler{
    . . .
    // Kết hợp thông điệp vào file bin của đối tượng
private fun combinerMessageIntoBinFile(message: String, binFile:
ByteArray, mode: ModeEncytion = ModeEncytion.LOWBIT): ByteArray{
    val tempBinFile = binFile.copyOf()
    val byteArrayMessage = convertStrToByteArray(message)
    val binFileSize = tempBinFile.size
    val bit = if(mode == ModeEncytion.HIGHBIT) 1 else 4
    for (index in byteArrayMessage.indices){
        tempBinFile[binFileSize-bit-index*bit] =
tempBinFile[binFileSize-bit-index*bit] xor byteArrayMessage[index]
    }return tempBinFile}
// Trích rút thông điệp nguyên mẫu từ file bin đã nhúng tin
fun getOriginalMessage(binFileOriginal: ByteArray, binFileEncrypted:
ByteArray): String{
    val byteArrayMessage: ArrayList<Byte> = ArrayList()
    if(binFileOriginal.size == binFileEncrypted.size){
        binFileOriginal.forEachIndexed { index, byte ->
            if(binFileEncrypted[index] != byte){
                byteArrayMessage.add(binFileEncrypted[index] xor byte)
            }
        }
    }
    return
convertByteArrayToStr(byteArrayMessage.toByteArray().reversedArray())
    // Giấu thông điệp vào file bin
fun hidingMessage(gltfFile: File?, binFile: File?, message: String,
```

```

mode: ModeEncyption): ByteArray?{
    var afterCombine: ByteArray? = null
    if(gltfFile != null){
        val jsonFileString = gltfFile.bufferedReader().use {
it.readText()} val mGLTFHandled =
Gson().fromJson<GLTFModel.GLTfComponents>(jsonFileString,
GLTFModel.GLTfComponents::class.java)
if(binFile != null){// Combine message into bin file
afterCombine =
combinerMessageIntoBinFile(message, binFile.readBytes(), mode)}}
    // Xuất file bin chứa thông điệp mật
    return afterCombine}}
. . .

```

2. Các màn hình tương tác / Activity

Màn hình chính

```

class MainActivity : AppCompatActivity() {
. . .
    // Khai báo Layout với FragmentManager
    arFragment = supportFragmentManager.findFragmentById(R.id.arFragment)
    as ArFragment
    arFragment.arSceneView.scene.addOnUpdateListener
    {arFragment.onUpdate(it)}
    // Nếu người dùng chưa đăng nhập, chuyển sang SignIn Activity
    if(auth.currentUser == null){
startActivity(Intent(this, SignInActivity::class.java)); finish()
}else{ // Lấy thông tin người dùng
    // Kiểm tra tài khoản đã đăng ký chưa
    database.child(userPathTree).addListenerForSingleValueEvent(object :
ValueEventListener{
    override fun onDataChange(snapshot: DataSnapshot) {
        if(!snapshot.hasChildren()){
            val key = database.child(userPathTree).push().key

```

```

        val user = User(key, mUsername, mGoogleAccount, mPhotoUrl)
        database.child(userPathTree).setValue(user)}}
        override fun onCancelled(snapshot: DatabaseError) {
            Log.i(TAG, "This account has been registered")}}}
// Kiểm tra sự kiện của Lớp User
database.child(userPathTree).addValueEventListener(object :
ValueEventListener{
    override fun onCancelled(snapshot: DatabaseError) {}
    override fun onDataChange(snapshot: DataSnapshot) {
        val dataUser = snapshot.getValue(User::class.java)
        if(dataUser!=null){try{
            if(!dataUser.binMessageUrl.isNullOrEmpty()){
storage.getReferenceFromUrl(dataUser.binMessageUrl.toString()).getFil
e(binInternal)}}catch (e: IOException){}}
            // Hàm đặt vật thể 3D trong không gian thực
placeObject(this@MainActivity, Uri.parse(gltfInternal.path),
MODEL_SCALE)
            // Thiết lập sự kiện cho bề mặt AR
            arFragment.setOnTapArPlaneListener { hitResult, __, __ ->
                val anchor = hitResult.createAnchor()
                if(renderModel == null) placeObject(this@MainActivity,
Uri.parse(modelLink), MODEL_SCALE)
                addNodeToScene(anchor, arFragment, renderModel)
                val originalMessage =
gltfHandler.getOriginalMessage(binInternal.readBytes(),
binFile.readBytes())
                Toast.makeText(this@MainActivity, "Message:
$originalMessage", Toast.LENGTH_LONG).show() . . . }
// Khai báo đối tượng
fun placeObject(context: Context, uri: Uri, scale: Float){
ModelRenderable.builder().setSource(context,
RenderableSource.builder().setSource(context, uri,
RenderableSource.SourceType.GLTF2)
.setRecenterMode(RenderableSource.RecenterMode.ROOT).setScale(scale)

```

```

.build()).setRegistryId(uri).build().thenAccept { renderModel = it }
.exceptionally {}

    // Thêm đối tượng vừa thiết lập vào màn hình
    private fun addNodeToScene(anchor: Anchor, arFragment:
ArFragment, renderable: ModelRenderable?){
        val anchorNode = AnchorNode(anchor)
        anchorNode.setParent(arFragment.arSceneView.scene)
        // Tạo một nút chuyển động cho phép biến đổi vật thể
        val node = TransformableNode(arFragment.transformationSystem)
        node.setParent(anchorNode)
        node.renderable = renderable
        node.select()}

    . . .

```

Màn hình đăng nhập tài khoản

```

class SignInActivity : AppCompatActivity(), View.OnClickListener {
    . . .
    companion object{
        private const val TAG = "#SignInActivity"
        private const val RC_SIGN_IN = 9001}

        // Cấu hình Google SignIn (GoogleSignInOptions)
        val gso =
        GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build()// Áp dụng cấu hình SignIn Options cho client
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso)}
        override fun onClick(v: View) {when(v.id){R.id.sign_in_button ->
signIn()}}

        private fun signIn() {
            val signInIntent = mGoogleSignInClient.signInIntent
            startActivityForResult(signInIntent, RC_SIGN_IN)
        }
    }
}

```

```
        // Xác thực với Firebase bởi tài khoản Google
private fun firebaseAuthWithGoogle(idToken: String?) {
    // Lấy chứng chỉ từ tài khoản
        val credential =
GoogleAuthProvider.getCredential(idToken, null)
        // Xác thực với Firebase
        mFirebaseAuth.signInWithCredential(credential)
            .addOnCompleteListener(this){
                if(it.isSuccessful){
                    startActivity(Intent(this, MainActivity::class.java))
                        finish()}}}
    . . .
```