Table 1: RASUES security analysis (Sniffing).

Sniffing						
Phase	**Where**	**Threat**	**Consequences**	**Attacker**	**Property**	**Mitigation**
1.1	Attester ↔ Relying party	Attacker may attempt to **listen** to the communication	The communication authenticity is broken among all entities	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance)	Secure TLS
1.2	Attester ← Relying party	Attacker may attempt to **listen** to the communication	Learning when the AWDT timer is initialized	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_2^S$ (Recoverability)	Secure TLS
1.3	Target environment → Attestation environment	Attacker may attempt to **listen** to the communication	Learning when the AWDT timer is invoked	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_2^S$ (Recoverability)	Secure TLS
1.4	RVP ← Relying party	Attacker may attempt to **listen** to the communication	Learning when the AWDT timer is started	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance)	Secure TLS
1.5	Verifier ← RVP	Attacker may attempt to **listen** to the communication	Learning the derived expected state in the whitelist	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_1^S$ (Updateability)	Secure TLS
1.6	Attester ← RVP	Attacker may attempt to **listen** to the communication	Learning the content of the update transmission	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_1^S$ (Updateability)	Secure TLS
1.7	Target environment → Attestation environment (2)	Attacker may attempt to **listen** to the communication	Learning the content of the update during update	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_1^S$ (Updateability)	Secure TLS
2.1	Attester← Verifier	Attacker may attempt to **listen** to the communication	Learning the content of the attestation request	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
2.2	Target environment → Attestation environment (3)	Attacker may attempt to **listen** to the communication	Learning the integrity measurement process	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
2.3	Target environment ← Attestation environment (2)	Attacker may attempt to **listen** to the communication	Learning the integrity measurement completed	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
2.4	Attester → Verifier	Attacker may attempt to **listen** to the communication	Learning the evidence through the attestation response, thus results in reducing its trustworthiness	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
3.1	Attester ← Verifier (2)	Attacker may attempt to **listen** to the communication	Learning the attestation result	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
3.2	Attester → Relying party	Attacker may attempt to **listen** to the communication	Learning the attestation result	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
3.3	Attester ← Relying party (2)	Attacker may attempt to **listen** to the communication	Learning the decision made by the relying party and knowing the deferral tickets	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_3^S$ (Freshness)	Secure TLS
3.4	Target environment → Attestation environment (4)	Attacker may attempt to **listen** to the communication	Learning the decision made by the relying party and knowing the deferral tickets	$\mathcal{A}_N$(Network)	$\boldsymbol{F}_4^S$(Integrity Assurance), $\boldsymbol{F}_2^S$ (Recoverability)	Secure TLS

Table 2: RASUES security analysis (Blocking).

Blocking						
Phase	**Where**	**Threat**	**Consequences**	**Attacker**	**Property**	**Mitigation**
1.1	Attester ↔ Relying party	Preventing entities from communicating with each other	The communication authenticity is broken among all entities	$\mathcal{A}_N$(Network)	F_4^S(Integrity Assurance)	TLS can detect violations
1.2	Attester ← Relying party	Attacker may attempt to **block access** to the entity	Blocking the **instruction of setting up the AWDT**	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	We assumed to be trusted
1.3	Target environment → Attestation environment	Attacker may attempt to **block access** to the entity	Blocking **invoking AWDT**	$\mathcal{A}_N$(Network)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	TPM in the attestation environment
1.4	RVP ← Relying party	Attacker may attempt to **block access** to the entity	Blocking **starting AWDT**	$\mathcal{A}_N$(Network)	F_4^S(Integrity Assurance)	Secure TLS
1.5	Verifier ← RVP	Attacker may attempt to **block access** to the entity	Blocking deriving expected state from whitelist	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS
1.6	Attester ← RVP	Attacker may attempt to **block access** to the entity	Update will not be delivered due to the RVP being blocked. Thus the attester will not be updated, and remain in its current state, either vulnerable or benign	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_1^S (Updateability)	RVP is Assumed to be trusted, since the timer has been initiated recently, an attack will be detected when the watchdog has timed out
1.7	Target environment → Attestation environment (2)	Attacker may attempt to **block access** to the entity	Blocking triggering update at the measurement	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_1^S (Updateability)	TPM in attestation environment
2.1	Attester← Verifier	Attacker may attempt to **block access** to the entity	Verifier not capable of initiating attestation request, thus cannot determine the trustworthiness of attester's expected state	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	IMA & cryptographic key exchange
2.2	Target environment → Attestation environment (3)	Attacker may attempt to **block access** to the entity	The integrity measurement will not be triggered at the attestation environment	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	IMA & cryptographic key exchange
2.3	Target environment ← Attestation environment (2)	Attacker may attempt to **block access** to the entity	The attestation environment will not provide the evidence of the measurement	$\mathcal{A}_N$(Network). $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	IMA & cryptographic key exchange
2.4	Attester → Verifier	Attacker may attempt to **block access** to the entity	The attester is unavailable to comply with required measurement, thus cannot continue with verification	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	IMA & cryptographic key exchange
3.1	Attester ← Verifier (2)	Attacker may attempt to **block access** to the entity	The effect is minimal, but the verifier cannot send the verified evidence to the attester	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	IMA & cryptographic key exchange
3.2	Attester → Relying party	Attacker may attempt to **block access** to the entity	The Appraisal process will fail due to the relying party not receiving the attestation result. Thus, the attester might be malicious or vulnerable. The device will not be able to receive any further updates and becomes completely unavailable for updates or further assessments or verification	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Out of scope
3.3	Attester ← Relying party (2)	Attacker may attempt to **block access** to the entity	Availability. Relying party is assumed to be trusted. The attester will be rest upon the watchdog timer has timed out	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	AWDT
3.4	Target environment → Attestation environment (4)	Attacker may attempt to **block access** to the entity	Blocking the decision made by the relying party	$\mathcal{A}_N$(Network), $\mathcal{A}_S$ (Software)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	AWDT

Table 3: RASUES security analysis (modifying).

Modifying						
Phase	**Where**	**Threat**	**Consequences**	**Attacker**	**Property**	**Mitigation**
1.1	Attester ↔ Relying party	Attacker may attempt to **modify the content** of the entity	Leads to impersonation of another malicious device	$\mathcal{A}_N$(Network), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance)	Secure TLS
1.2	Attester ← Relying party	Attacker may attempt to **modify the content** of the entity	Attacker modifying the AWDT timer. Preventing the attester from resetting if it malicious	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Cryptographic key exchange
1.3	Target environment → Attestation environment	Attacker may attempt to **modify the content** of the entity	Modifying the invoke request, so the attestation environment might be invoked when it is not supposed to be, compromising integrity	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Cryptographic key exchange, Secure TLS, TPM
1.4	RVP ← Relying party	Attacker may attempt to **modify the content** of the entity	Modifying the content of the AWDT, so that attacker misconfigure the initializing time	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance)	Cryptographic key exchange, Secure TLS, AWDT
1.5	Verifier ← RVP	Attacker may attempt to **modify the content** of the entity	Modifying the whitelist	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, IMA
1.6	Attester ← RVP	Attacker may attempt to **modify the content** of the entity	Modifying the supplied update before installation	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software). $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, TPM
1.7	Target environment → Attestation environment (2)	Attacker may attempt to **modify the content** of the entity	Modifying the attestation environment or triggering measurement by unintended entity	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, TPM
2.1	Attester← Verifier	Attacker may attempt to **modify the content** of the entity	Invoking untrusted or modified attestation challenge requests	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Secure TLS, IMA, Cryptographic key exchange
2.2	Target environment → Attestation environment (3)	Attacker may attempt to **modify the content** of the entity	Learning the integrity measurement process	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
2.3	Target environment ← Attestation environment (2)	Attacker may attempt to **modify the content** of the entity	Modifying the integrity measurement of attestation	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
2.4	Attester → Verifier	Attacker may attempt to **modify the content** of the entity	Modifying the evidence. Verifying wrong evidence or wrong device, resulting in an attester to be legitimate Note: what is included in the evidence, such as device ID, once, public/private keys, and PCR	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
3.1	Attester ← Verifier (2)	Attacker may attempt to **modify the content** of the entity	Modifying the attestation results provided by the verifier. This breaks the integrity of RATS passport-model	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM
3.2	Attester → Relying party	Attacker may attempt to **modify the content** of the entity	Attacker modifying the evidence and the results to be sent to the relying party. The replying party might be fooled to accept the given evidence, thus making a wrong decision	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM
3.3	Attester ← Relying party (2)	Attacker may attempt to **modify the content** of the entity	Modifying the decision made by the relying party and changing the deferral tickets of AWDT. Preventing it from resetting on time	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Secure TLS, AWDT
3.4	Target environment → Attestation environment (4)	Attacker may attempt to **modify the content** of the entity	Modifying the decision and AWDT by the attester	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Secure TLS, AWDT

Table 4: RASUES security analysis (replay/spoof).

Replay/Spoof						
Phase	**Where**	**Threat**	**Consequences**	**Attacker**	**Property**	**Mitigation**
1.1	Attester ↔ Relying party	Attacker may attempt to replay or spoof the communication	Untrusted entities will participate in the communication, thus replay or spoof other entities, compromising integrity	$\mathcal{A}_N$(Network), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance)	Secure TLS
1.2	Attester ← Relying party	Attacker may attempt to replay or spoof the initialization of AWDT using new or existing credentials	The attester will comply with malicious AWDT initialization, thus it will become under attacker's control	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Cryptographic key exchange
1.3	Target environment → Attestation environment	Attetester replay or spoof AWDT invoking process	The attester will starts with incorrect AWDT initialization, thus manipulated	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Cryptographic key exchange, Secure TLS, TPM
1.4	RVP ← Relying party	Attacker incorrectly informing the relying party about the AWDT initialization	The relying party will miscalculate the timer and become under the attacker control	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance)	Cryptographic key exchange, Secure TLS, AWDT
1.5	Verifier ← RVP	Spoofing the expected state from the RVP	Attester can always mark itself as legitimate after remote attestation	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, IMA, Cryptographic key exchange
1.6	Attester ← RVP	Installing malicious update, and impersonating the RVP	This led the attester to be controlled by the attacker	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, TPM
1.7	Target environment → Attestation environment (2)	Spoofing attester to trigger measurement	The attester will perform the measurement	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_1^S (Updateability)	Secure TLS, TPM
2.1	Attester← Verifier	Attacker impersonating verifier and initializing attestation request	The attester will comply repeatedly to an untrusted verifier, thus, verifier has unauthorized access to attesters' data	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Secure TLS, IMA
2.2	Target environment → Attestation environment (3)	Attacker impersonate target environment to invoke attestation environment	This cause the attestation environment to replay to untrusted	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
2.3	Target environment ← Attestation environment (2)	-	-	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
2.4	Attester → Verifier	-	Using existing communication credentials in the attester or spoof it and reply to the verifier	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM, IMA
3.1	Attester ← Verifier (2)	-	-	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM
3.2	Attester → Relying party	-	The attester can reproduce evidence/results and send it to relying party to be appraised	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Cryptographic key exchange, TPM
3.3	Attester ← Relying party (2)	-	The attacker can send a wrong decision / deferral ticket to the attester. Resulting in always trusting the atester	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_3^S (Freshness)	Secure TLS, AWDT
3.4	Target environment → Attestation environment (4)	-	-	$\mathcal{A}_N$(Network), $\mathcal{A}_S$(Software), $\mathcal{A}_P$(Privileged)	F_4^S(Integrity Assurance), F_2^S (Recoverability)	Secure TLS, AWDT